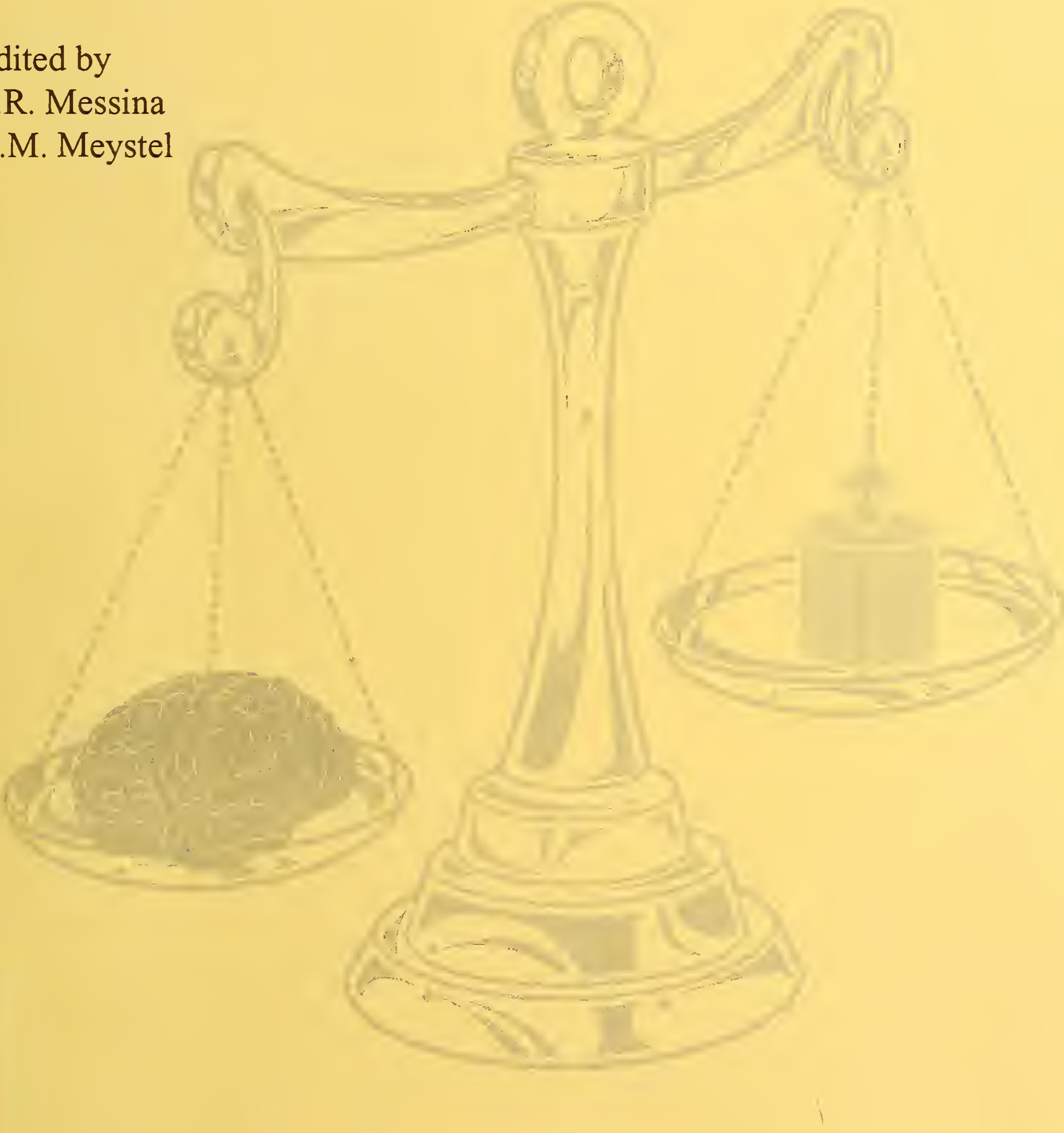
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NIST Special Publication 990

Measuring the Performance and Intelligence of Systems: Proceedings of the 2002 PerMIS Workshop August 13 - 15, 2002

Edited by
E.R. Messina
A.M. Meystel



The National Institute of Standards and Technology was established in 1988 by Congress to "assist industry in the development of technology . . . needed to improve product quality, to modernize manufacturing processes, to ensure product reliability . . . and to facilitate rapid commercialization . . . of products based on new scientific discoveries."

NIST, originally founded as the National Bureau of Standards in 1901, works to strengthen U.S. industry's competitiveness; advance science and engineering; and improve public health, safety, and the environment. One of the agency's basic functions is to develop, maintain, and retain custody of the national standards of measurement, and provide the means and methods for comparing standards used in science, engineering, manufacturing, commerce, industry, and education with the standards adopted or recognized by the Federal Government.

As an agency of the U.S. Commerce Department's Technology Administration, NIST conducts basic and applied research in the physical sciences and engineering, and develops measurement techniques, test methods, standards, and related services. The Institute does generic and precompetitive work on new and advanced technologies. NIST's research facilities are located at Gaithersburg, MD 20899, and at Boulder, CO 80303. Major technical operating units and their principal activities are listed below. For more information visit the NIST Web site at http://www.nist.gov, or contact the Publications and Program Inquiries Desk, 301-975-3058.

Office of the Director
- National Quality Program
- International and Academic Affairs

Technology Services
- Standards Services
- Technology Partnerships
- Measurement Service
- Information Services
- Weights and Measures

Advanced Technology Program
- Economic Assessment
- Information Technology and Applications
- Chemistry and Life Sciences
- Electronics and Photonics Technology

Manufacturing Extension Partnership Program
- Regional Programs
- National Programs
- Program Development

Electronics and Electrical Engineering Laboratory
- Microelectronics
- Law Enforcement Standards
- Electricity
- Semiconductor Electronics
- Radio-Frequency Technology [1]
- Electromagnetic Technology[1]
- Optoelectronics[1]
- Magnetic Technology[1]

Manufacturing Engineering Laboratory
- Precision Engineering
- Manufacturing Metrology
- Intelligent Systems
- Fabrication Technology
- Manufacturing Systems Integration

Chemical Science and Technology Laboratory
- Biotechnology
- Process Measurements
- Surface and Microanalysis Science
- Physical and Chemical Properties[2]
- Analytical Chemistry

Physics Laboratory
- Electron and Optical Physics
- Atomic Physics
- Optical Technology
- Ionizing Radiation
- Time and Frequency[1]
- Quantum Physics[1]

Materials Science and Engineering Laboratory
- Intelligent Processing of Materials
- Ceramics
- Materials Reliability[1]
- Polymers
- Metallurgy
- NIST Center for Neutron Research

Building and Fire Research Laboratory
- Applied Economics
- Materials and Construction Research
- Building Environment
- Fire Research

Information Technology Laboratory
- Mathematical and Computational Sciences[2]
- Advanced Network Technologies
- Computer Security
- Information Access
- Convergent Information Systems
- Information Services and Computing
- Software Diagnostics and Conformance Testing
- Statistical Engineering

---

[1]At Boulder, CO 80303.
[2]Some elements at Boulder, CO.

NIST Special Publication 990

Measuring the Performance and Intelligence of Systems: Proceedings of the 2002 PerMIS Workshop August 13–15, 2002

Edited by:
E.R. Messina
A.M. Meystel
Intelligent Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230

Co-Sponsored by:
National Institute of Standards and Technology
Defense Advanced Research Projects Agency
Institute of Electrical and Electronic Engineers Control Systems Society
National Aeronautics and Space Administration

In Cooperation With:
Institute of Electrical and Electronic Engineers Neural Network Council

Issued September 2002

U.S. Department of Commerce
Donald L. Evans, Secretary

Technology Administration
Phillip J. Bond, Under Secretary of Commerce for Technology

National Institute of Standards and Technology
Arden L. Bement, Director

National Institute of Standards and Technology Special Publication 990
Natl. Inst. Stand. Technol. Spec. Publ. 990, 442 pages (September 2002)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 2002

For sale by the Superintendent of Documents, U.S. Government Printing Office
Internet: bookstore.gpo.gov — Phone: (202) 512-1800 — Fax: (202) 512-2250
Mail: Stop SSOP, Washington, DC 20402-0001

TABLE OF CONTENTS

PART I
RESEARCH PAPERS

PART I
RESEARCH PAPERS

1M1 - Performance Metrics

1. Performance Metrics for Intelligent Systems: An Engineering Perspective
   R. Gao, Purdue University
   L. Tsoukalas, Purdue University

2. RCS Based Hardware-in-the-loop Intelligent System Design and Performance Measurement
   S. Ananthakrishnan, Pathway Technologies
   S. Agrawal, University of Delaware
   R. Venugopal, Pathway Technologies, Inc.
   M. Demeri, Ford Research Lab

3. Evaluating the Performance of Assistive Robotic Systems
   H. Yanco, University of Massachusetts, Lowell

4. Measuring Classifier Intelligence
   J. DeLeo, National Institutes of Health

5. Problems of Performance Measurement in Locally-Organized Systems
   V. Stefanuk, Russian Academy of Science

Performance Metrics for Intelligent Systems
An Engineering Perspective

Rong Gao and Lefteri H. Tsoukalas
Purdue University
School of Nuclear Engineering
Applied Intelligent Systems Laboratory
W. Lafayette, IN 47907-1290
gao@ecn.purdue.edu, tsoukala@ecn.purdue.edu

ABSTRACT

This paper provides a general discussion on performance metrics for intelligent systems drawing largely on our experience with engineering applications. The experience has led us to view machine intelligence as a type of machine-facilitated human intelligence. This view implies that the locus of machine intelligence is to be found in relations amongst humans *vis a vis* the machine (in the subject-to-subject-via-object relation). Hence, quantitative metrics for intelligence may be sought as functions of the human-machine and machine-machine interface; evaluating them may be achieved through a conventional behaviorist-type of approach where a system is characterized by observing its response to given inputs. Some guidelines for this process that we found to be useful are discussed in the paper.

Keywords: *Intelligence, Performance Metrics, Fuzzy Logic, Neural Networks*

1. INTRODUCTION

Since the onset of industrial revolution machines have been developed to relieve humans from tedious work and reduce the cost of producing goods and services. With the advent of the digital computer, the industrial notion of machines as artifacts capable of mechanical work has been greatly expanded to include sophisticated capabilities of information processing, decision-making, communications, sensing, coordination and control. Somewhat naively but, as we will argue latter on, quite accurately, we tend to refer to this post-industrial sophistication as "machine intelligence."

Equipped with control and computing units, machines can be made to be very predictable and quite reliable. The reliability is exemplified by the fact that machines follow given instructions literally regardless of changes in their environment. This characteristic is undesirable in complex applications. Predefined instructions are usually coded by human experts and make it nearly impossible to take into account every possible scenario that might happen in real-world applications. Therefore, it is desirable that machines possess capabilities for appropriately handling cases that they have not been directly instructed. Although there have been numerous definitions of intelligence, engineers tend to think that a machine or a system in general is intelligent if it is capable of handling exceptions properly. In this respect the engineering view is somewhat similar to that of the cognitive scientists who look at various *agnosias* (for example, *visual agnosia* or failure to recognize objects seen) as opportunities and tools for understanding the complex inner-workings of the human brain (Gazzaniga, 1998). But, the engineering aim is quite different. Intelligent systems are transforming the way we design, fabricate, operate and even dispose complex engineering artifacts such as airplanes and power plants. A Boeing 777 airliner, for example, may have in excess of nine million distinct parts, while the number for an advanced boiling water reactor is the 109 range. Intelligent systems are necessary to make such systems safe, economical and manageable at all times.

It should be noted that the adjective "intelligent" gradually fades away from designating any system that becomes routinely available and widely familiar. Intelligence is an attribution reserved for systems that are at a more nascent level of development and more likely not proven or established technology. Building intelligent systems is a goal that often appears to be quite elusive. Hence, having a metric for intelligence, any metric, is useful not only for comparing system A to system B, but also, for comparing system A at an early age of development to system A at more mature level of development. In this respect an index for intelligence is not different from any metric of performance that can be consistently applied to assess the growth of a system. This a very important issue for systems such as nuclear power plants or passenger airliners whose lifespan may be comparable or exceeding the lifespan of their designers and operators.

We have observed in numerous engineering applications that a major difficulty giving rise to the elusiveness of machine intelligence is due to our deeply held notions and assumptions about machines. Humans seem to be so overwhelmingly prepared to think of machines as quite independent entities, separate from us; distinct and also distinguished; sitting outside the boundaries of human boundaries (physiologically, cognitively, and socially); and yet so intimately ours. Machines are always adjuncts to humans. Although we describe them in terms of objective qualities (such as power, mass, volume) their most important attributes are the ones relating to their functionality and purpose (interfacial characteristics). Intelligence is a functionality not an objective property. Yet, and because of that, machine intelligence involves what the psychologists call *reification*, that is, something appears to exist just because we have word(s) for it.

The view we espouse is that machines, including the sophisticated and computationally savvy artifacts of today and tomorrow, are accessories to human intelligence. They have no intelligence of their own (to have that they would have to live the lives of humans). Intelligent systems are machines functioning as a medium for playing out the drama of human intelligence; principally activities for asking and answering questions, a kind of generalized dialogue amongst humans (not anymore constrained to be physically present). Our view defines intelligent systems as *virtual interlocutors*, that is as systems that function in a way that makes it possible for humans and other machines to ask and answer questions unconstrained by personal presence or awareness. In this sense, machines can be viewed as "intelligent" although we all know that they could not possibly come about on their own, without human volition, know-how, design and material implementation.

Viewing intelligent machines as "virtual interlocutors," raises the question of language. What is the right idiom for the man-machine discourse we are talking about? It has to be a language that its ultimate aim is to facilitate a virtual dialogue amongst humans and as such it is desirable to have the computational characteristics of natural language. For this, we have to turn to fuzzy logic. It is extremely difficult to capture within any formal language the complex and rich attributes of natural language including, but not limited to, flexibility, semantic depth, computational economy (parsimony), and portability. We strongly believe that fuzzy logic is a highly promising tool; its potential is largely uncapped, its full power is still to be harnessed.

Additional frustrations with intelligent machines are caused by the lack of a bridge that links any interpretation of intelligence (such as the one put forward in this paper) to the implementation of intelligence. Definitions of intelligence do not often provide useful information needed by engineers in the realization process. Engineers would like to have quantitative performance metrics that could be used to measure the degree of intelligence of a system. Defining a performance metric is, however, not easier than building an intelligent system itself. Theoretically, provided a quantitative performance metric is available, methods can be developed to optimize the system's performance to reach a threshold that this particular system is deemed intelligent. We have gathered plenty of experience in designing optimization algorithms and they are being used in a variety of applications involving neural networks and fuzzy logic (Tsoukalas, 1997). The research on performance metrics for intelligent systems is an important focus. In the following sections we present a general discussion and some guidelines for designing performance metrics.

2. Intelligent Systems

Researchers of artificial intelligence have traditionally defined intelligence as an inherent property of a machine. An intelligent machine (or system) is viewed as one that has some computational capacity to act like a human, that is, "think" humanly, or "act" rationally, or "think" rationally (Russel, 1985). Hence, computational metrics of intelligence are traditionally expected to measure how well a machine performs like a human, for example, like a chess master, or like an expert diagnostician.

We believe that such thinking-like-human approaches trying to mimic the way a mind operates are not technically feasible with current engineering capabilities. First, the complexities of human brain make simulation impossible using existing technology. The latest Intel Pentium IV processor integrates 55 million transistors, much less than the 100 billion neurons and 100 trillion synaptic junctions found in a person's brain. Second, the basic processing unit in a computer system, the transistor, is identical throughout a processor and can only handle two numbers, 0 and 1. On the other hand, neurons are diversified and are capable of processing subtle electrochemical signals efficiently in numerous possible ways. Third, humans as complex biological systems are the result of millions of years of natural selection. Many species coexist but only humans have emerged as intelligent creatures (in a full sense of the word). And certainly part of their story is found in language and their capacity to form complex cultural and technical artifacts and social institutions.

We think of intelligence as an advanced functionality of a system. Based on the discussion above it is quite clear that this type of functionality should be independent of a machine's internal implementation details. We propose that this functionality is a function of the interface (human-machine, machine-machine); to be found in the subject-to-subject relation *vis a vis* the object, that is, the computer or "intelligent machine." In order for the intelligent functionality to be something observable (measurable), the intelligence of a system ought to be judged based on a system's response to provided inputs. The internal structure and implementation may be not so important for the purpose of observing it.

But, is it really possible to measure intelligence quantitatively? We believe that the answer ought to be yes; else, we run the risk of viewing machine intelligence as something metaphysical, mysterious and therefore not amenable to investigation. Methodologically, we know that it is very hard to quantify machine intelligence. The reason is that any intelligence measure has to be an overall performance index involving many detailed measures. For example, an intelligent person may be extraordinarily good at math but very poor in music. For another person the opposite may be true (capable in music but incapable at math). Obviously there is no single number that can be used to characterize the difference between these two persons. Human IQ tests have been criticized for their non-typicality, unreliability and inconsistency. The incommensurability of intelligence is a barrier we have to face with intelligent machines as well; and not only because they are used or they are better at different things. Evidently, it is different humans that make for very different intelligent machines.

Despite all these difficulties, defining performance metrics for intelligent systems is a worthy goal. People realize that the long lack of universally acceptable measures has seriously hampered the process of intelligent systems development. Science and technology have advanced by cooperation and competition. The root of cooperation and competition is a common ground with which results of different researchers can be compared. Comparisons are impossible without agreement on performance metrics.

Some philosophical and methodological barriers to intelligent metrics can be overcome by adopting a pragmatic approach. Such an approach focuses on the specifics of the problem and calls for strategies for improvement (learning) and development (maturation) within a given context and with well-defined metrics. Thus, for the first step a pragmatic approach is to identify the needs of the applications. What are the situations that intelligent systems are designed for? Are we going to develop an intelligent system that clones a human being (in some way) or solve a specific problem (for the benefit of well-defined user needs)? Undoubtedly, the efforts involved in designing performance metrics for these two different systems are rather incomparable.

3. General Guidelines

In this section we present five items we found important in designing performance metrics. They are intended to be general guidelines. A comprehensive performance metric should take into account all five of the proposed items. The first describes a questionnaire-based method similar to approaches taken for knowledge solicitation in well-defined application domains. The second identifies the capacity for generalization. The third item identifies the need for adaptation. The fourth captures the social or group capabilities of intelligent systems. Finally, the fourth identifies transitivity or how different intelligent systems ought to be comparable.

3.1 General and Specific Metrics

Detailed quantitative metrics of general intelligence are difficult to formulate and potentially not necessary. Intelligence in general integrates so many parameters and is not possible to have an objective general measure. However, approximate and application oriented measures are possible. It is a lot easier to develop a metric to evaluate the performance of some system for a specific application like chess playing or medical diagnosis. Therefore, if possible, application-specific measures should be always considered first. Application-specific measures can be constructed based on a **set of questionnaires.** Techniques from knowledge solicitation and web-based assessments can be used. Questionnaires can be analyzed via statistical approaches or fuzzy quantification (Tsoukalas, 1997).

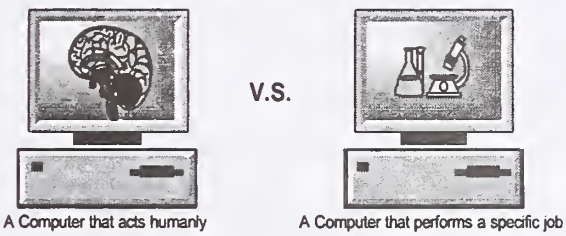


Figure 1. General intelligence and specific intelligence

3.2 Metric for Generalization Capabilities

The degree of intelligence ought to reflect in some fashion the capacity for generalization. An intelligent system solves a problem first by searching its previous experience for similar cases. The first level of intelligence is looking for a direct match. A higher level of intelligence is needed when a direct match is unavailable. The higher level of intelligence appears as the capability of maneuvering experience in part in order to generate new unseen instances, which resemble the problem to solve, as shown in figure 2. Suppose, for example, that the problem is to classify some unknown shape. The direct match is the first approach and essentially compares the given shape against ideal geometrical shapes. The indirect match may do the matching against generalized instances and more generally against composite generalized instances. An appropriate metric in this example ought to capture the ability of the system to deal with the more general topological transformations involved in the indirect matches. Although a lot has been written for generalization, typically generalization in many engineering applications is little

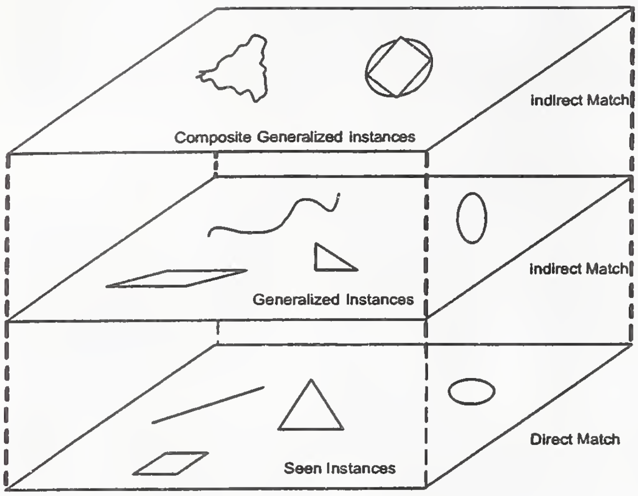


Figure 2. An illustration of different levels of generalizations

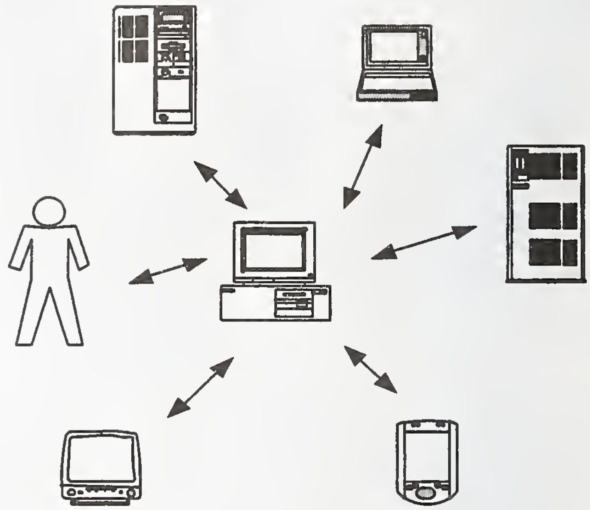


Figure 3: System and its environment

different from interpolation. But, that's fine. Even a good interpolation metric is adequate and very useful if applied consistently as a measure of generalization.

3.3 Intelligence Metric should be Adaptive

The intelligence of a system cannot really be evaluated by a fixed rule. Rather it ought to be a collective index that reflects the overall performance of the evaluated system on a variety of situations. Consequently, this metric should be dynamic and adaptive. It should change to adapt to the new information that has been gathered regarding a system's more recent performance. In this respect a simple neural network can be very useful as means of adaptation of the metric (even though the range of adaptation may be rather narrow).

3.4 Intelligence as a Social Characteristic

From an engineering viewpoint, an isolated system, no matter how intelligent it is, is not of great interest. Measuring intelligence should be performed in the context of a group or society that includes other systems (computers or humans), as shown in figure 3. Any intelligent ability ought to be evaluated from the interrelations among multiple systems. In a sense the kind of machine intelligence we are called to quantify almost always involves network systems, be they computers, sensors, robotic devices, controllers, expert systems, or search engines in the Internet. The criteria for judging the social abilities of interacting systems are the correctness of interpretations of their inputs and the effectiveness of presentation of the outputs.

If a system is put into a society, we have to consider its relations with other parties. In a society, a system always stores in its local database the profiles of other systems that it knows. In order words, the intelligence profile of a system is distributed to the society. The argument that an intelligence metric should be dynamic and adaptive requires that a system be constantly monitored. Assigning a dedicated agent to perform this job will be biased and unreliable. The solution is that any system ought to be examined by its peers (the rest of the systems in the same society). This implies that an intelligent system (human or computers) should posses the ability to evaluate the intelligence of other systems. However, it is impossible for one system to evaluate all other systems directly, partly because of security reasons or simply because of the exponentially growing communication overhead. In such cases, one system needs to reach its own judgment indirectly based on the judgment (which may be direct or indirect one) of other systems. For example, in figure 4, System 1 is about to evaluate the performance of System 3. However, System 1 has no means to communicate with System 3 directly. In such a scenario, it is possible for System 1 to reach a decision based on the evaluations obtained from System 2 and System 4 (with which it has direct connections). The third party information might be direct (such as System 2 that has direct connection with System 3) or indirect (such as System 4, which in return relies on System 5 to make its own decision). To achieve this type of feature, a system needs to "trust" to some degree the capabilities of other systems.

3.5 Transitivity

Finally any intelligence metric should be used with care, especially when comparisons are involved. The direct comparison between two systems using an intelligence metric

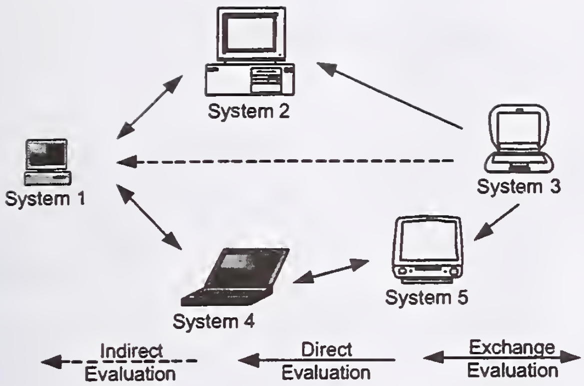


Figure 4. Indirect evaluation of intelligence

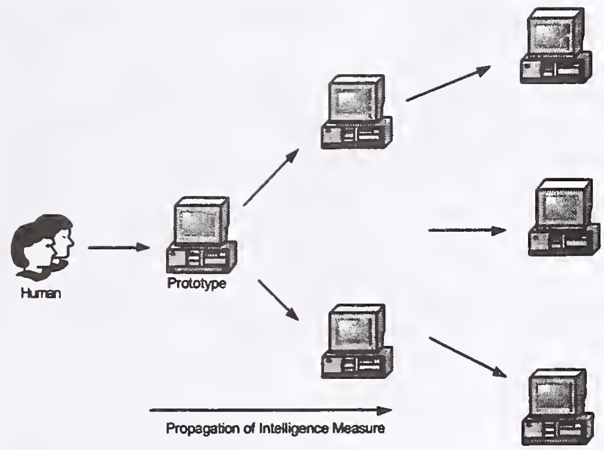


Figure 5. Construct intelligence metric in a networking approach

remains questionable, except when this metric has been defined in a very narrow sense such as for the performance of a specific job such as medical diagnosis. Generally, we should avoid the use of an intelligence metric in a chained fashion because of its inherent uncertainty and incommensurability. For instance, system A is more intelligent than system A, which is more intelligent than system C. Yet, it is not correct to imply that system A is more intelligent than C.

4. A Prototype for General Measures

An intelligence metric in general is difficult to be written in an analytical form. However, an engineering construction approach may be useful. The process of evaluating intelligence itself is an intelligent process. To break this infinite loop, we must start from some systems that are canonically intelligent. Humans are the main option. Some prototype machine systems are first constructed and are approved to be intelligent by humans. These initial systems are not necessarily perfect in terms of natural intelligence. The criteria of intelligence are numerous and the most important one is the capacity of judging the intelligence of other systems. The intelligence evaluation process is not a calibration process where a less precise machine is able to calibrate a more precise one. A more pragmatic approach is needed. The prototypes that have been evaluated by humans now can evaluate other machines, as shown in figure 5. The evaluated systems can be further used to evaluate other systems. It should be noted that this is not a one shot operation and the first round of evaluation is usually inaccurate. In later iterations, every system (including the first prototype) will have the chance to improve its evaluation capability by comparing others' evaluations with its own one. A steady state for the system, if reachable, informs us that it has achieved stable and more accurate evaluations for other systems. The key of this approach is that an intelligent system is able to talk to its neighbors.

5. Conclusions and Remarks

We have discussed several important guidelines for intelligent metrics. The key is to focus on the interface (machine-machine, machine-human). A system's intelligence is reflected by the ways of processing inputs and presenting outputs. The interface not only accepts problem-solving data but only those "control" or "judgment" pieces of information, such as evaluations by other systems.

Over the years there has been a great interest in constructing intelligent systems not only because machines can potentially solve problem more consistently and flexibly (with human supervision) but also because intelligent systems are a great metaphor for intelligent human activities. The activities of posing and answering questions and of building knowledge through a dialectic process are now greatly facilitated by computer systems which we tend to view as "intelligent." The results are an unprecedented and much needed access to the human mind. And, machines that not only surprise and fascinate us but, most importantly, we cannot do without them; in the sense that we cannot really manage the complexity of exceeding complicated engineering artifacts that come to be over several generations.

However, there is no free lunch. The associated cost is the so called *responsibility dilemma*. A system is intelligent because it is capable of handling exceptions that it was never taught. The question then may be raised as to who should be responsible for the consequence incurred by the "intelligent actions." The designer of the system should not be blamed because the system does not follow instructions literally and the designer can not foresee the direction that the system evolves. These issues have to be addressed and if possible reflected in future intelligence metris.

6. References

Gazzaniga, M.S., Ivry, R.B., Mangun, G.R., *Cognitive Neuroscience: The Biology of the Mind*, W.W. Norton & Company, New York, 1998.

Tsoukalas, L. H, and Uhrig, R. E., *Fuzzy and Neural Approaches in Engineering*, John Wiley and Sons, NY, 1997.

Russell, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall, New Jersey, 1995

7. Biography

Rong Gao obtained his bachelor's and master's degrees from Tsinghua University, Beijing in 1994 and 1997 respectively. In December 2000, he received his master degree in electrical and computer engineering, Ph.D. in nuclear engineering, both from Purdue University. He is currently working at the Artificial Intelligence Systems Laboratory (AISL) as a visiting Assistant Professor. His research interests include human-machine interface, intelligent signal processing and complex systems modeling.

Lefteri H. Tsoukalas is Professor and Head of the School of Nuclear Engineering at Purdue University and the Director of the Applied Intelligent Systems Laboratory (AISL). He has a PhD from the University of Illinois (1989) and numerous publications in neurofuzzy systems including the book *Fuzzy and Neural Approaches in Engineering*, co-authored with R.E. Uhrig (Wiley, 1997).

RCS Based Hardware-in-the-loop Intelligent System Design and Performance Measurement

Ananthakrishnan S., Chief Engineer, Pathway Tech. Inc., Blue Bell, PA

Agrawal, Sunil, Professor, Univ. of Delaware, Newark, DE

Venugopal, Ravinder, Consultant, Pathway Tech. Inc., Blue Bell, PA

Demeri, Mahmoud, Senior Tech. Specialist, Ford Research Lab, Dearborn, MI

Abstract:

In this paper, we present an approach to the design of intelligent systems based on RCS architecture that allows seamless transition from modeling and non real-time simulation to real-time simulation and subsequent hardware-in-the-loop testing. This methodology provides a unified, structured, hierarchical environment, so that "analytical design" of intelligence can be seamlessly transferred to machine/manufacturing process intelligence. As part of the research, we present two case studies wherein we demonstrate how RCS architecture and functionality can be incorporated using commercial software and hardware environment. The enhancements to commercial software in the areas of (i) knowledge hierarchy, (ii) open, modular, and structured programming using RCS architecture, (iii) minimal software programming, (iv) advanced control design methodologies, and (v) efficient numerical schemes for optimization provide a framework for comparing qualitative and quantitative measures of performance improvement over traditional industrial automation hardware that uses PID cards, programmable logic controllers (PLCs), and other microprocessor based controllers with limited functionality.

In order to compete in the global market place, engineering organizations are under increasing pressure to design, develop, and deploy products in the market place as quickly as possible with first time quality. In order to achieve these objectives, it is necessary to streamline the design and development process, namely, "transfer of analytical design of intelligence to mechatronics intelligence" in an efficient and expedient manner. Using Real-time Control System (RCS) architecture that organizes the elements of intelligence to create functional relationships and information flow across levels following principles of hierarchy and assigned responsibilities at each level [1, 2], we have implemented control systems for two applications, namely, a cable robot, and electrohydraulic test system.

The objective of this research effort was to conduct case studies on how RCS architecture can be used in a flexible automation scenario where traditional industrial control cards (hardware) do not provide adequate measures of performance. In addition, industrial control hardware and real-time software primarily focus on embedded software with no structured approach or methodologies for real-time simulation or hardware testing of intelligent system design. Further, in commercial software used for hardware-in-the-loop testing, there is a general lack of well-defined intelligent infrastructure. Hence, from the standpoint of intelligent system design and performance metrics, through these design examples, we demonstrate the need for a unified environment for design and development of intelligent systems that combine knowledge hierarchy, computational schemes, dynamic models, etc., so that platform configuration, and repetitive coding can be minimized [3].

Intelligent control has been a focus of attention for researchers over the past three decades. Initially, it was viewed as interaction of artificial intelligence and control systems [4]. Another major attempt to formalize the discipline of intelligent controls includes theories of nested hierarchical information structures to address control of complex systems [2, 5, 6]. Nonetheless, all these approaches emphasize the importance of "analytical design" of intelligent machines and focus of system functions pertaining to machine intelligence.

In summary, design of intelligent systems based on methodologies described in [2, 4, 5, 6] have the following three common characteristics: (i) utilization and implementation of concepts and ideas from diverse disciplines, (ii) "additional controllers" to accommodate intelligent system performance that utilize knowledge based techniques to meet performance requirements, (iii) emphasis on the overall system coordination and integration as opposed to control specific system components. In this paper, through the two case studies we demonstrate how commercial software can be adapted to meet these objectives of intelligent system design through hardware-in-the-loop testing. The qualitative measures of performance enhancement with this design approach are, (i) structured environment using RCS architecture (ii) system models for rapid plug and play design, (iii) minimal platform configuration and coding. Quantitative measures of performance enhancement are, (i) improved control design and implementation techniques over commercial control hardware, (ii) seamless migration from simulation to hardware testing, (iii) cost effective intelligent system design modules for commercial environment.

System Dynamics and Control

Cable Robot Dynamics and Control:

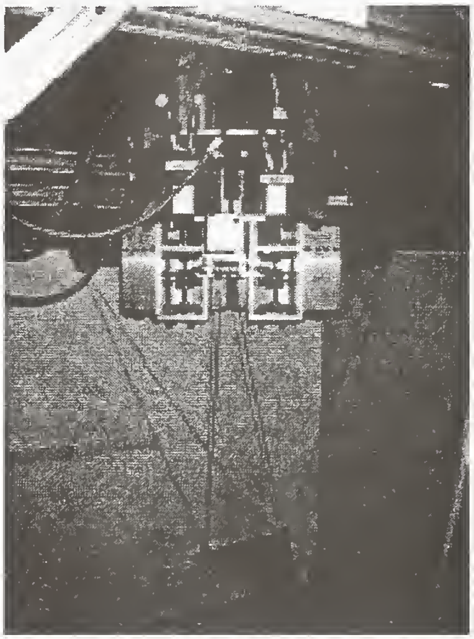


Figure 3. Cable Robotic System

In order to systematically explore the questions of configuration design, coordination, and intelligent control, we have implemented a hardware-in-the-loop control design environment for a cable robot as shown in Figure 3. The system consists of 6 degrees-of-freedom cable robot mounted on a two degree-of-freedom X-Y gantry structure. Cable suspended robots have one unique property – *they carry loads in tension but not in compression*. Due to this feature, well-known results in robotics for trajectory planning and control are not directly applicable to cable robots, but must be modified to reflect the constraints of positive cable tensions [8, 9]. In this paper, we present RCS based control implementation of design algorithms taking into consideration that the cables have to be in tension for effective control.

Based on Newton-Euler formulation, the equations of motion for the cable system without considering the gantry motion can be written as

$$\begin{bmatrix} m\ddot{x}_m \\ m\ddot{y}_m \\ m\ddot{z}_m - g \\ I\begin{pmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{pmatrix} + \begin{pmatrix}\omega_1\\ \omega_2\\ \omega_3\end{pmatrix} \times I \begin{pmatrix}\omega_1\\ \omega_2\\ \omega_3\end{pmatrix} \end{bmatrix} = -\tilde{J}^T(q)u \qquad \text{(i)}$$

where in equation (i), m is the mass matrix, I is the moment of inertia of the end-effector about its center of mass with respect to the basis vector $[b_1 \quad b_2 \quad b_3]$ [8]. The above equation can be rewritten in the following form

$$D(x)\ddot{x} + C(x,x)\dot{x} + G(x) = -\tilde{J}^T(q)u \qquad \text{(ii)}$$

where $x = [x_m \quad y_m \quad z_m \quad \psi \quad \theta \quad \phi]^T$ and it's derivatives refer to the configuration of the end-effector plate and q is a vector of cable lengths. The functional relation between ω_i, α_i and (ψ, θ, ϕ) is given by

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}=\begin{bmatrix}1 & 0 & -S\theta\\ 0 & C\psi & S\psi C\theta\\ 0 & -S\psi & C\psi C\theta\end{bmatrix}\begin{bmatrix}\dot{\psi}\\ \dot{\theta}\\ \dot{\phi}\end{bmatrix} \qquad \text{(iii)}$$

with $\alpha_i = \dot{\omega}_i$ and $G(x) = [0 \;\; 0 \;\; -mg \;\; 0 \;\; 0 \;\; 0]^T$.

Based on the dynamic equations described in equation (ii), a *Lyapunov based controller* was developed with the candidate Lyapunov function in equation (iv)

$$V(\tilde{x}, x) = \frac{1}{2}\dot{x}^T D(x)\dot{x} + \frac{1}{2}\tilde{x}^T K_P \tilde{x} \qquad \text{(iv)}$$

where $\tilde{x} = x - x^d$ with x^d as the reference trajectory. In order to ensure asymptotic stability of equation (ii) about x^d, a control law was developed in **[8]**, to ensure positive control. The control law given in equation (v)

$$\begin{bmatrix}\tilde{J}^T & \tilde{X} & \dot{\tilde{X}}\end{bmatrix}\begin{bmatrix}u\\ K_p\\ K_d\end{bmatrix} = G(x) \qquad \text{(v)}$$

has the structure of $Ay = b$ such that $y > 0$ with $\tilde{X}$ having a dimension of (6 X 6) with diagonal entries of $\tilde{x}_i$. The resulting optimization problem to ensure positive control results in minimizing $\|Ay - b\|_2$ and was solved using *lsqnonneq*, a nonnegative least squares problem solver. The computations of pseudo-inverse and Singular Value Decomposition (SVD) are directly applicable to least squares optimization problems. The technique of *lsqnonneg* is an iterative process. Consider a system of equations represented by $Ay = b$, the technique consists of computing the residue $R = b - Ay_e$, where y_e is the initial guess or estimate of the solution. The main process of *lsqnonneg* consists of iteratively computing $w = A^T R$, and improving the estimates y_e in each solution step. The process iterates till all elements of w are less than the specified tolerance, and the number of iterations reach an upper limit. In each iteration step, the pseudo-inverse of a subset of the elements of matrix A is used to compute the new estimates y_e. The pseudo-inverse is computed using SVD. As part of the knowledge hierarchy, the intelligent system design involved developing a real-time S-function C-code API structure in Matlab/Simulink that would allow simultaneous optimized gain selection using *lsqnoneg* and Lyapunov based controller. This was implemented using RCS methodology in the cable robotic system at University of Delaware. The details of RCS based implementation of this design and the results of this experimental study are discussed later.

Electrohydraulic Test System:

New tooling concepts and an advanced binder control unit with individually controlled hydraulic cylinders has recently been developed to allow the local control of metal flow into the die cavity during a stamping operation. Forces are applied on the sheet metal blank using a set of hydraulic cylinders mounted on the lower bolster of the press. In a hydraulic press, the ram depresses the piston of each one of the hydraulic cylinders in the binder area, thus compressing the hydraulic fluid and raising the pressure inside the cylinders. This pressure is transferred through the piston back up to the blank. To obtain the desired force on the blank, the pressure within the hydraulic cylinder is regulated by modulating the flow of hydraulic fluid out of the cylinder by means of a closed-loop control system. But, unlike a hydraulic press where the ram maintains a constant velocity profile, the piston velocity of the mechanical press is a nonlinear function of time, and therefore, the differential equation that relates rate of pressure increase to servo-valve opening is nonlinear. Hence, control of pressure within the cylinder cannot be achieved with simple PID control in the case of a mechanical press, although this would be possible in the case of a hydraulic press for which the piston velocity is constant during the stamping cycle. In this case, standard off-the-shelf PID cards were not able to control the cylinder pressure in a mechanical press.

In this case, we demonstrate an RCS based nonlinear controller **[10]** using a single cylinder test stand as shown in Figure 4. As can be seen, the hydraulic cylinder in the middle of the platform will be controlled as the table is actuated using the crank assembly using motors shown in Figure 4. The schematic of the hydraulic system consisting of reservoir, servovalve, and other hydraulic components is shown in Figure 5.

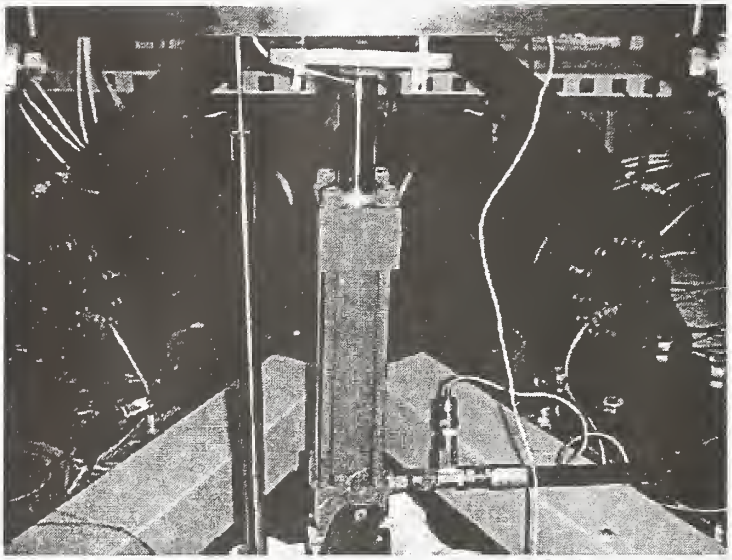


Figure 4. Hydraulic System Test Stand for Single Cylinder Test

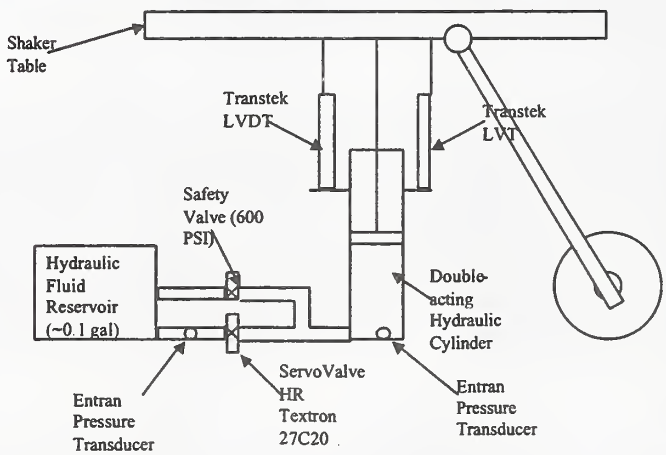


Figure 5. Schematic of the Hydraulic Test System for Closed Loop Control

System Dynamics:

Figure 6 shows the mechanical ram and the hydraulic force control unit in a typical mechanical press. The governing equations of motion for the system shown in Figure 6 consists of cylinder dynamics of the hydraulic system and the dynamics of the mechanical crank drive as described in the following equations (1) and (2):

$$\dot{P}(t) = \frac{\beta}{A(s - d(t) + \varepsilon)}\left(A\dot{d}(t) - \alpha(t) K_{servo} \sqrt{\frac{2(P(t) - P_t)}{\rho}} \right) \quad (1)$$

where $P(t)$ is the cylinder pressure, P_t is the tank pressure, A is the cross-sectional area of the cylinder, $d(t)$ is the displacement of the piston from top-dead-center (TDC), $0 \le \alpha(t) \le 1$ is the amount by which the servo-valve is

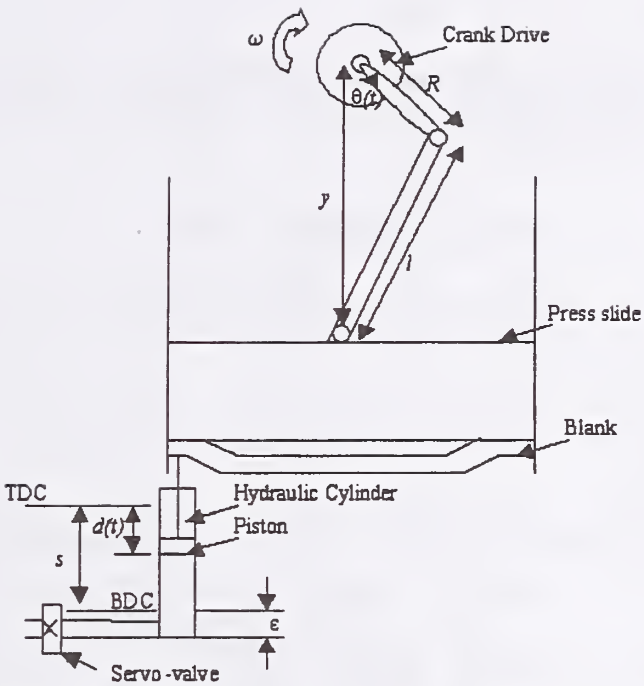


Figure 6. Schematic of Hydraulic Force Actuation in Mechanical Press

opened with $\alpha(t) = 0$ representing the valve fully closed and $\alpha(t) = 1$, ρ is the density of the hydraulic fluid, K_{servo} is the effective cross-sectional area of the valve orifice, s is the stroke length of the piston and ε is the height of fluid in the cylinder when the piston reaches bottom-dead-center (BDC) as shown in Figure 6. We now mathematically model a simple mechanical crank press drive to obtain a relationship between the displacement of the piston and its velocity, assuming that all the links is rigid as shown in equation (2)

$$d(t) = R\cos\theta(t) - \sqrt{l^2 - R^2 \sin^2\theta(t)} - R - l + s. \qquad (2)$$

where $\theta(t)$ is the crank angle, R is the crank radius, and l is the length of the coupler, as shown in Figure 7. Differentiating Equation (2) results in Equation (3) as shown below.

$$\dot{d}(t) = -R\sin\theta(t)\omega - \frac{R^2 \sin 2\theta(t)\omega}{2\sqrt{l^2 - R^2 \sin^2\theta(t)}}, \qquad (3)$$

The resulting nonlinear model of equation (1) and (3) describes the dynamics of the hydraulic actuation system used in a mechanical press. The nonlinear nature of the equations clearly indicates that standard PID control will not be sufficient for precise closed-loop pressure control.

Control System Design:

The controller design for the hydraulic force actuation unit is based upon equation (1) and (3) of the preceding section. The nonlinear control technique used for designing the controller is feedback linearization. Feedback linearization is used in the control of nonlinear systems in which the nonlinearity is known and invertible. It involves the use of additional terms in the control signal to cancel out the nonlinearity, after which a classical linear controller is used on the effectively linear system. Using the structure of the nonlinearity, we choose

$$\alpha(t) = \frac{A}{K_{servo}} \sqrt{\frac{\rho}{2(P(t) - P_t)}} \left(\dot{d}(t) - \frac{(s - d(t) + \varepsilon)}{\beta} v(t) \right) \qquad (4)$$

where $v(t)$ is a linear control term to be determined subsequently. Substituting equation (1) into equation (4) results in

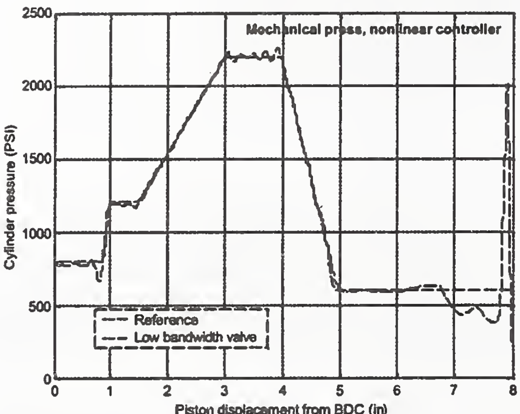
$$\dot{P}(t) = v(t) \tag{5}$$

Thus, the control law in equation (4) converts the nonlinear control problem in equation (1) into a first order linear ODE involving the new control variable *v(t)*. The first type of controller that we choose for the system given by (5) is a PI controller of the form shown in equation (6), namely,

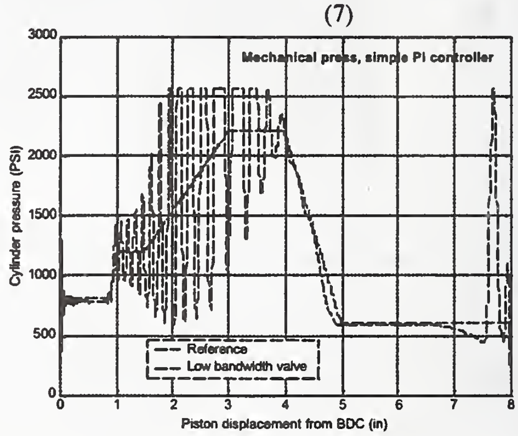
$$v(t) = K_p(P_{ref}(t) - P(t)) + K_i \int_{t_{0i}}^{t} (P_{ref}(\sigma) - P(\sigma))d\sigma, \tag{6}$$

where K_p is the proportional gain, $P_{ref}(t)$ is the desired pressure command, and K_i is the integral gain. Substitution of equation (6) into equation (4) results in the following nonlinear control law,

$$\alpha(t) = \frac{A}{K_{servo}}\sqrt{\frac{\rho}{2(P(t)-P_t)}}\left(\dot{d}(t) - \frac{(s-d(t)+\varepsilon)}{\beta}(K_p(P_{ref}(t)-P(t)) + K_i\int_{t_{0i}}^{t}(P_{ref}(\sigma)-P(\sigma))d\sigma)\right) \tag{7}$$



Nonlinear Control Simulation for a Mechanical Press



PI Control Simulation for a Mechanical Press

Figure 7. Cylinder Pressure Comparison between Linear and Nonlinear Control

Figure 7 shows the comparison between pressure profiles using linear and nonlinear controllers. The pressure profiles are given from bottom dead center (BDC) of the cylinder, and so must be viewed from right to left. As the ram comes down and makes contact with the cylinder, the initial pressure spike to the right is noticed. Then, as the ram plunges the cylinder, the nonlinear control is able to track the reference pressure trajectory, while the PI control causes significant oscillations. So, a simple change from hydraulic ram to a mechanical ram rendered the conventional control cards inadequate for effective control. The structure of nonlinear control used in this design can be generalized to a class of systems. Essentially, since the structure of nonlinearity is known in pressure control of hydraulic systems, we can use feedback linearization techniques.

A second approach to control of the cylinder pressure in the binder force control unit use estimates of plant dynamics in a pole assignment controller leading to pole assignment adaptive control. A deterministic auto regressive moving average (DARMA) model of the plant represented by equation (8) represents the plant dynamics

$$A(q^{-1})y(t) = q^{-d}B'(q^{-1})u(t) \tag{8}$$

where the coefficients of the polynomial $A(q^{-1}), B'(q^{-1})$ are given by equation (9) and (10)

$$A(q^{-1}) = a_0 + a_1q^{-1} + \ldots + a_{n1}q^{-n1} \tag{9}$$

$$B(q^{-1}) = (b_0 + b_1 q^{-1} + ... + b_{m1} q^{-m1}) q^{-d} \quad (10)$$

The plant dynamics can be estimated based on past measurements of $y(t), u(t)$ by rewriting equation (8) in normalized form where $a_0 = 1$ as follows:

$$y(t) = \phi(t-1)^T \theta \quad (11)$$

with

$$\phi(t-1)^T = [-y(t-1), -y(t-2), ..., u(t-1), u(t-2)...]$$
$$\theta = [a_1, a_2, ..., b_1, b_2, ...]$$

The parameters of the system can be estimated using recursive least squares (RLS) algorithm and its variation such as RLS with covariance resetting as follows:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-2)\phi(t-1)}{1+\phi(t-1)^T P(t-2)\phi(t-1)}(y(t) - \phi(t-1)^T \hat{\theta}(t-1)) \quad (12)$$

$$P(t-1) = P(t-2) - \frac{P(t-2)\phi(t-1)\phi(t-1)^T P(t-2)}{1+\phi(t-1)^T P(t-2)\phi(t-1)} \quad (13)$$

For covariance resetting, equation (13) is reset periodically by a known value of covariance as follows

$$P(t_i - 1) = k_i I \quad (14)$$

The control input $u(t)$ in pole assignment controller is determined by solving the following equation

$$\hat{A}(t,q^{-1})\hat{L}(t,q^{-1}) + \hat{B}(t,q^{-1})\hat{P}(t,q^{-1}) = A^*(q^{-1}) \quad (15)$$

where $\hat{\theta}(t)$ consists of estimates $\hat{A}(t,q^{-1}), \hat{B}(t,q^{-1})$ and $\hat{L}(t,q^{-1}), \hat{P}(t,q^{-1})$ are unique polynomials of order $(k-1)$ and $A^*(q^{-1})$ is the desired polynomial selected based on intended closed loop behavior. The feedback control law that combines the estimates of the plant dynamics and equation (15) is given by [12, 13]

$$\hat{L}(t,q^{-1})u(t) = \hat{M}(t)y^*(t) - \hat{P}(t,q^{-1})y(t) \quad (16)$$

where

$$\hat{M}(t) = \frac{1 + a_1^* + ...}{\hat{b}_1 + ...} \quad (17)$$

with $y^*(t)$ same as $P_{ref}(t)$, $y(t)$ same as $P(t)$ and $u(t)$ same as the valve command similar to the nonlinear controller.

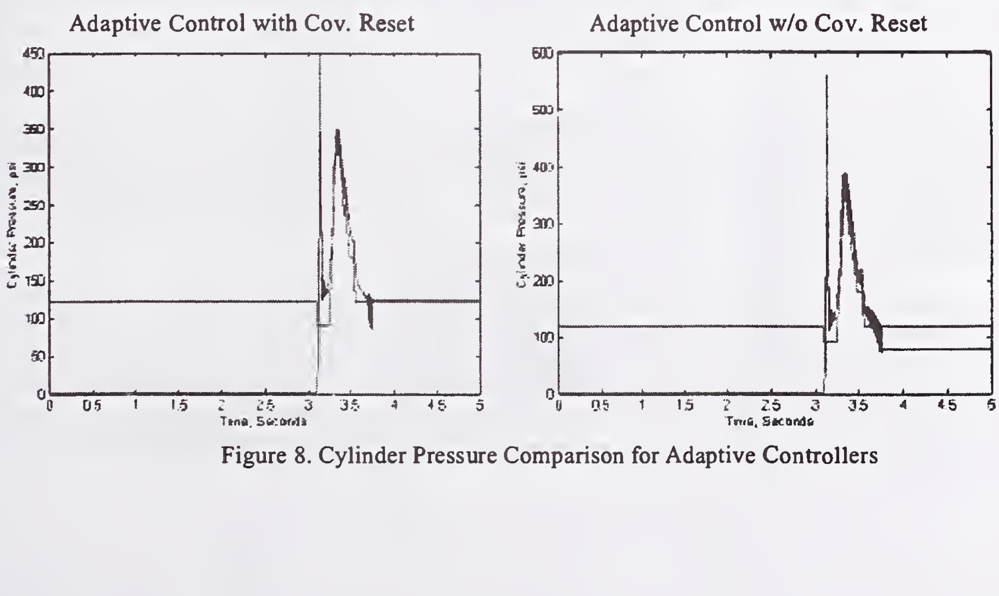


Figure 8. Cylinder Pressure Comparison for Adaptive Controllers

Figure 8 shows the comparison between pressure profiles using adaptive controllers with or without covariance reset. As can be seen, unlike, linear controllers, both the adaptive controllers show a stable response like the nonlinear controller. But, in the case of adaptive controller with covariance resetting, in addition to the tracking, the response is well behaved in the presence of plant and measurement noise, similar to the nonlinear controller. The initial spike in the response is due to the initial transient as the ram comes down and plunges on the cylinder. So, from these analytical results, it is clear that the nonlinearities in the system require linear control structure with adaptation or a nonlinear controller in order to get desired response behavior.

RCS Implementation:

In a goal driven, sensory interactive, and system behavior based intelligent architecture, we have used the four key paradigms and four key elements **[1, 2]** as guideline for design of intelligent systems. With simulation as the heart of the development process whether it be non real-time or real-time or hardware-in-the-loop, our case studies support the entire intelligent design process without having to transfer data, change design environment, or write extensive custom code. Further, the design process allows models of physical systems, namely, kinematics and dynamics of cable robot and mechanical press test fixture, to be used for non real-time simulation and subsequent hardware-in-the-loop testing of various real-time control schemes, namely, linear, nonlinear, adaptive, and Lyapunov based controllers. In order to avoid repetitive coding we have developed application programmers' interfaces (APIs) that allow optimized integrated code generation, and embedded system options that allow seamless transition of intelligent controllers from simulation to hardware testing.

Figure 9 shows the functional decomposition of the information flow for the cable robotic system. As can be seen from the schematic, the design flow uses the four key paradigms, and four key elements of RCS control node to decompose the design problem into a multi-layered hierarchical control problem. The tension sensor signals {TS1,...,TS6} are fed back to the lowest layer of the hierarchy to close tension feedback control loops using control laws that receive tension request from the Prim process and the filtered sensor data from the sensory processing {SP1, ..., SP6}. Based on whether the tension request in within the desired threshold, the value judgment {VJ1, ...,VJ6} is used to limit the tension request so that appropriate motor commands are generated by the behavior generation module {BJ1, ..., BJ6}. The motor command to the servo drives that actuate the motor are thus assured to be within operational limits in order to achieve effective tension control.

The commands to the *tension control RCS nodes* are generated by the *Prim process*, which also uses the elements of the RCS node to implement the *Lyapunov controller* in order to ensure that positive control is achieved. The optimization problem to ensure positive control results in minimizing $\|Ay-b\|_2$ and was solved using *lsqnonneq*, a nonnegative least squares problem solver. *Since the basic Simulink structure does not support real-time optimization and control, S-function C-code API structure had to be developed as part of the knowledge hierarchy to demonstrate seamless transition from simulation to hardware testing.* As part of this knowledge hierarchy, the computations of pseudo-inverse and Singular Value Decomposition (SVD) are directly applicable to least squares optimization problems and hence the associated algorithms were also developed. As discussed earlier, at each control step the lyapunov controller requires the solution of *lsqnonneq*, which consists of iteratively computing $w = A^T R$, and improving the estimates y_e in each solution step. The process iterates till all elements of w are less than the specified tolerance, and the number of iterations have an upper limit. In each iteration step, the pseudo-inverse of a subset of the elements of matrix A is used to compute the new estimates y_e. The pseudo-inverse is computed using SVD. The Prim process, based on desired reference trajectories x^d from the E-move process computes the tension request u of equation (v) to the *tension control RCS nodes* based on this Lyapunov controller. In this structure of the Prim process, *lsqnonneq* serves as the VJ module for the *Lyapunov controller* that acts as a BJ module based on desired and actual trajectories that were generated using encoder feedback and associated SP computations that involves computation of *Jacobian* at each control step.

Similar methodologies were used in design of controllers for the electrohydraulic test system used for stamping process automation prototyping and for the sake of brevity not discussed here. In summary, the analytical methodologies, knowledge hierarchy and algorithms developed for estimation, optimization and control of intelligent systems provide a framework for software and associated real-time control development that will allow seamless transition from simulation to hardware. Figure 10 shows the framework of software developed for the case studies. As can be seen, this process can be streamlined for generalized development of knowledge hierarchy and algorithms to allow simulation and hardware testing of intelligent systems. Figure 11 shows the response plots and GUI used in real-time testing of cable robot. As can be seen, the *Lyapunov controller* works very well while maintaining positive tensions. Similar GUI was developed for design and testing of the electrohydraulic test system.

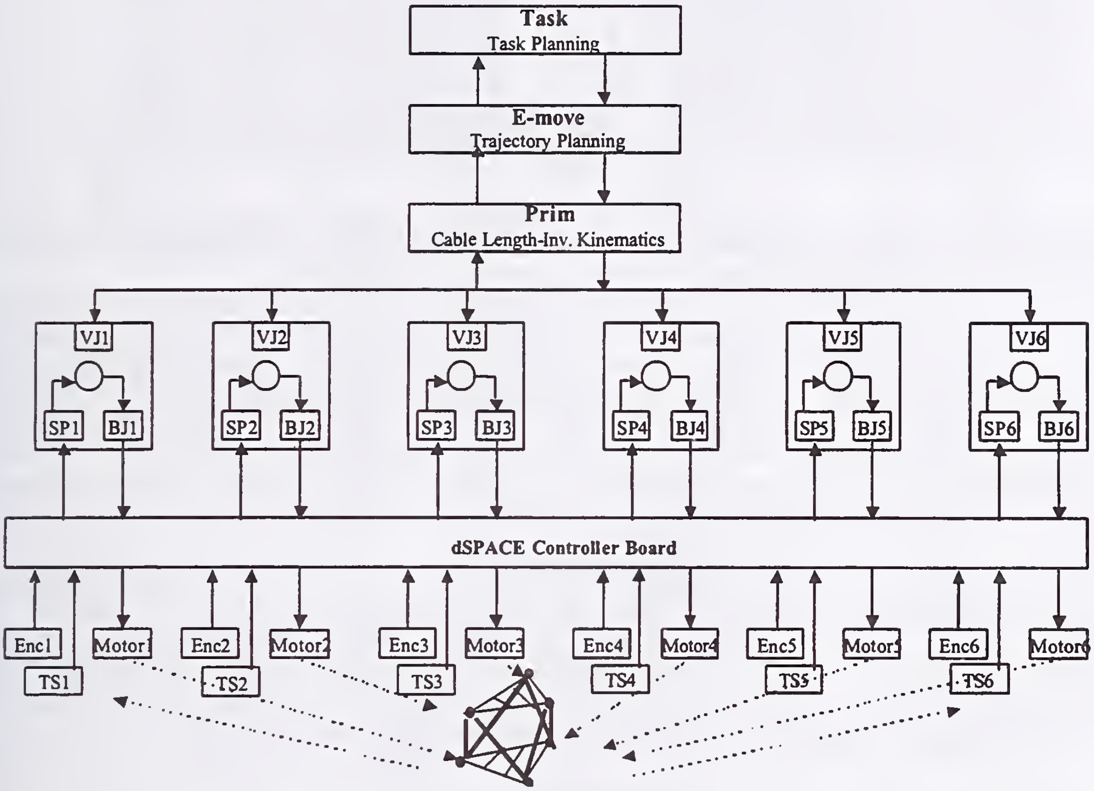


Figure 9. Hierarchical Decomposition of Intelligent Control Design for Cable Robot

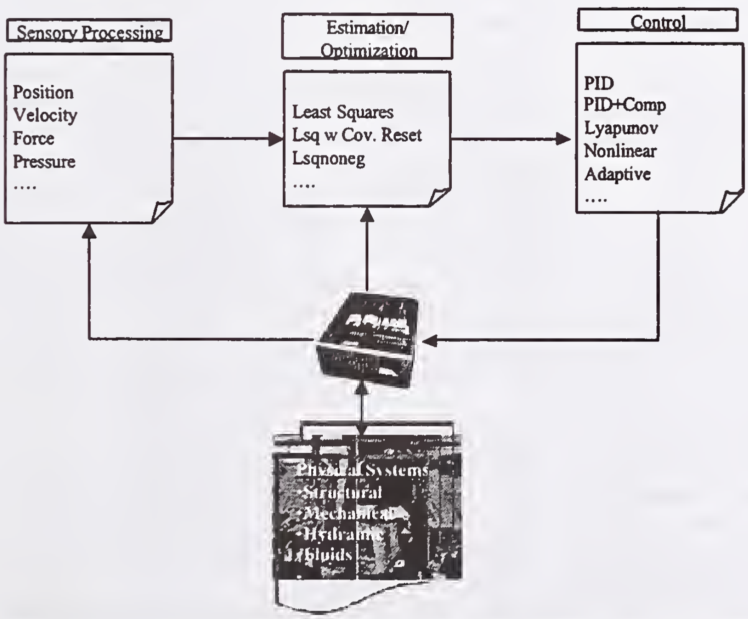


Figure 10. Real-time Estimation, Control, Optimization and System Model Software

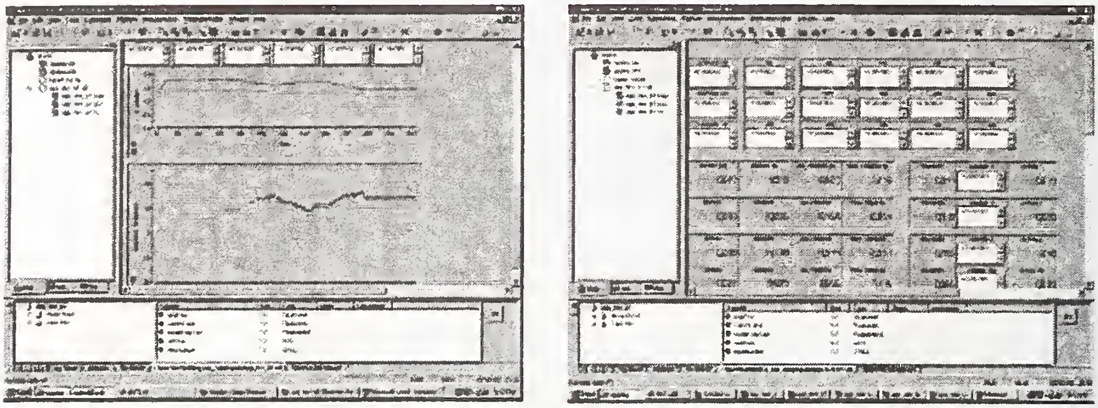


Figure 11. Response Plots and GUI for Real-time Testing of Cable Robot

In summary, we present research and industrial case *studies that allow efficient design and development of intelligent systems using RCS architecture in order to provide a unified, structured, hierarchical environment so that a designer can build software and associated real-time control without undue focus on software programming and hardware interface during the development process. This design approach allows real-time production code generation of the intelligent control design in a cost effective manner, thus providing improvement over commercial PIDs and PLCs. Rapid plug and play design and seamless migration from non real-time simulation to hardware testing provide qualitative measures of perfomance enhancement while the improved system behavior through advanced control provide quantitative measures of performance improvement.*

Acknowledgements

The authors wish to acknowledge that this work was done with funding from NIST, USAMP/AMD Project 301, and corporate funding from Pathway. We specifically, wish to thank Elena Messina, Harry Scott, Huang Hui-Min, and Jim Albus of NIST, Dr. Mahmoud Demeri of Ford for their valuable comments during this project. In addition, we thank Soreyok Oh and Dr. Viswanath Ananth for their help in conducting the experiments. The Mechanical System Laboratory of University of Delaware under the direction of Sunil K. Agrawal developed the *Cable Robot* hardware for this project.

References

[1] Huang, H., Scott, H., Messina, E., Juberts, M., and Quintero, R., "Intelligent System Control: A Unified Approach and Applications", NIST Report, 1999.

[2] Albus, J. S.," Outline for the Theory of Intelligence," *IEEE Transactions on System, Man, and Cybernetics*, Vol. 21, No.3, 1991, pp 473-509.

[3] Ananthakrishnan, S., Agrawal, S.K., " Intelligent Software and Animation Tools for RCS Architecture Based Development – Phase I Final Report", NIST SBIR Project #276, February 2001.

[4] Fu, K.S., "Learning Control Systems and Intelligent Control Systems: An Interaction of Artificial Intelligence and Automatic Control", IEEE Transactions on Automatic Control, 1971, 70-72.

[5] Saridis, G.N., " *Self Organizing Control of Stochastic Systems* ", Marcel Dekker, New York, 1977.

[6] Meystel, A.M., "Intelligent Control in Robotics", *Journal of Intelligent and Robotic Systems*, Vol. 5, No. 4, 1988.

[7] Agrawal, S.K., Gardner, G., Pledgie, S., " Design and Fabrication of a Gravity Balanced Planar Mechanism Using Auxiliary Parallelograms", *Journal of Mechanical Design, Transactions of ASME*, Vo. 123, No. 4, 2001, 525 – 528.

[8] Basar, A. A., Agrawal, S. K., " Cable Suspended Robots: Feedback Controllers with Positive Inputs", Proceedings of American Control Conference, 2002.

[9] Basar, A.A., Agrawal, S. K., "Cable Suspended Robots: Design, Motion Planning, and Control", Proceedings of IEEE Conference on Robotics and Automation, 2002.

[10] Venugopal, R., Ananthakrishnan, S., " Closed Loop Simulation Study of Binder Force Control Unit", Pathway Technologies Incorporated, Report to USAMP Project 301, 2002.

[11] Venugopal, R., Ananthakrishnan, S., " Open Loop Simulation Study of Binder Force Control Unit", Pathway Technologies Incorporated, Report to USAMP Project 301, 2002.

[12] Fullmer, R., Ananthakrishnan, S., "Experimental Adaptive Controllers for a Class of Electrohydraulic Servo Systems", *Int. Journal of Adaptive Control and Signal Processing*, Vol. 7, 1993.

[13] Ananthakrishnan, S., et al., " Role of Estimation in Contact Dynamics Enhancement of Space Station Automated Integration and Assembly Facility, *IEEE Robotics and Automation Magazine*, Vol. 3, No. 3, 1996.

Evaluating the Performance of Assistive Robotic Systems

Holly A. Yanco
Computer Science Department
University of Massachusetts Lowell
One University Avenue
Lowell, MA 01854
holly@cs.uml.edu

ABSTRACT

When designing robotic systems for the disabled, it is necessary to demonstrate that the systems are safe. Beyond safety, it must also be shown that the equipment improves the quality of life for its user. This paper discusses methods for testing assistive robotic systems to assure safety, usability and usefulness. An example of a user test designed for a robotic wheelchair is presented and discussed.

KEYWORDS: *Assistive technology, assistive robotics, robotic wheelchair, user tests*

1 INTRODUCTION

Assistive technology enables people to do things that would be impossible without the technology. Assistive technology can range from smart homes for the elderly to robotic wheelchairs to voice control software for a computer. Robotic workstations can provide people with vocational assistance [Dallaway et al 1995, Kazi et al 1998, Wagner et al 1999]. Robotic walkers can be used to allow elderly people with decreased vision to walk around their nursing home [Lacey 1999].

In this paper, we discuss the testing process for assistive mobile robots which either carry or lead their users, so that the user and the robot must travel together during the use of the assistive system. We will call these devices assistive navigation systems. We will use Wheelesley, a robotic wheelchair system developed for indoor and outdoor use, in our discussions to illustrate the evaluation methods [Yanco 2000].

The target community for a robotic wheelchair system consists of people who are unable to drive a powered wheelchair using a standard joystick. This group includes people with cerebral palsy, stroke patients who omit stimuli from one side, and quadriplegics. The users vary in ability and access methods[1] used to drive the wheelchair. Some people can move a joystick, but are unable to make fine movement corrections using the joystick. Other people are able to click one or more switches using their head or other body part. Some potential users are unable to control a powered wheelchair with any of the available access devices and must rely upon a caregiver to move them throughout the world. A robotic wheelchair will enable this population to better self-navigate through the world, increasing independence.

Human-robot interaction must be considered when designing assistive travel systems. Designing a poor interface will result in an unusable system. Robotic wheelchairs must be able to connect to a variety of commonly used access methods. User tests must utilize the access methods to be used by the target population, even when testing able-bodied subjects. With a target population lacking the fine motor control necessary to move a joystick, user tests with able-bodied subjects using a joystick can not be extrapolated to the intended users.

2 EVALUATORS FOR SYSTEMS

When designing and evaluating assistive navigation systems, there are three groups of people that should be involved: providers, able-bodied test subjects and people in the target population of the system.

Providers are the people who prescribe and deliver systems to a user. In the case of wheelchairs, physical therapists adapt a wheelchair to its user by creating custom cushions, determining the proper access method, and adjusting settings such as speed controls. Physical therapists also work with wheelchair users to teach them how to use the system. Since they are very involved with providing care to users, it is important to involve these providers from the early stages of development through the final product testing. People who work individually and daily with users will have an understanding of the needs of the population.

Systems should be tested first on able-bodied subjects. Many members of the target population for robotic wheelchairs are non-verbal, making it difficult to do user tests since it is important to be able to tell if the user feels comfortable when testing the system. Walkers for the elderly infirm who have limited vision should first be tested on people who can see. The move to a target user should be made only after the safety and reliability of the system has been repeatedly demonstrated through able-bodied user tests.

[1] An access method is a means for controlling a powered wheelchair, such as a joystick or a sip-and-puff system.

3 EVALUATION OF SYSTEMS

The performance of an assistive navigation system must be measured through user tests. Tests should range from preliminary demonstrations of safety and usability to long term use by one or two subjects from the target population.

Before testing the system on users from the target population, tests on a large number of able-bodied subjects should be undertaken. These tests usually involve a test course, where subjects must traverse the course using assisted navigation and unassisted navigation for comparison. Metrics to be collected include time required to travel the course, number of safety violations which may range from scrapes to bumper hits to more serious failures, and the amount of effort required to drive the course.

A system's performance should be measured by conducting user tests that compare the performance of the robotic system against a non-robotic solution. For example, a robotic wheelchair can be designed to allow it to be controlled with no sensor mediation (manual control) or with sensor-mediated (robotic) control. By designing a system this way, the same user interface and access methods can be used to compare user performance with assisted control and with unassisted control. The human-robot interaction must be duplicated even when the robotic control is not used for navigation. It would be impossible to directly compare a traditional powered wheelchair driven by a joystick to a robotic system driven with an access method such as single switch scanning. This difference is even more problematic due to the fact that the target community does not have the fine motor control necessary to drive a powered wheelchair using a joystick. Care must be taken to design user tests that change only a small number of variables (ideally one).

Tests involving the target users should focus on long term use instead of runs on a test course. Before these long term tests of a small group of target users, researchers should consider undertaking long term tests of a few able-bodied subjects so that the test methods may be fully evaluated before proceeding to test subjects with reduced mobility.

4 ABLE-BODIED USER TEST DESIGN

To illustrate the design of able-bodied user tests, this section discusses indoor user tests designed for and performed with the Wheelesley robotic wheelchair system [Yanco and Gips 1998]. An experiment to test the performance of able-bodied subjects under robotic assisted control and under standard manual control was designed to determine if robotic assistance improved driving performance using single switch scanning as an access method. Single switch scanning is the access method of last resort for powered wheelchairs, primarily because drift is a significant problem. To correct a drift to the left or the right, the user must stop going forward, wait for the scanning device to get to the arrow for the direction of choice, click to turn the chair, stop turning, wait to scan to forward and then click to move forward again.

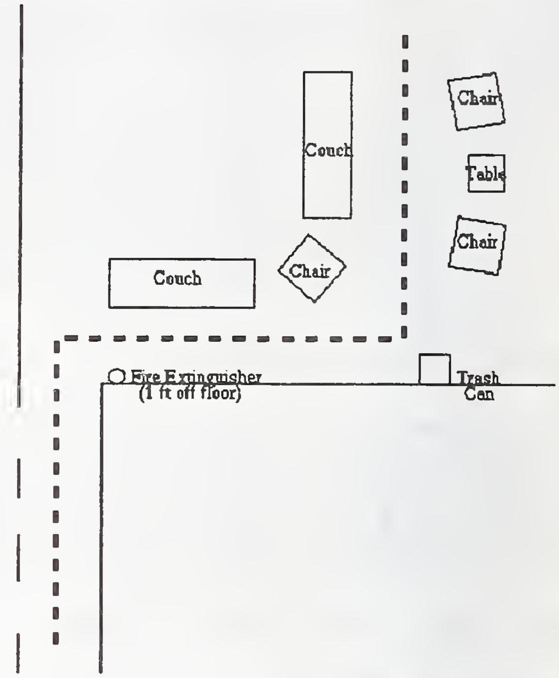


Figure 1: The indoor test course. A test run consisted of an up and back traversal of this course.

Fourteen able-bodied subjects (7 men and 7 women), ranging in age from 18 to 43, were tested. All subjects were familiar with using computers and none had driven the wheelchair before.

At the beginning of a session, the subject was shown the wheelchair. Sensors that are used in robotic assisted control were pointed out and explained briefly. Safety measures, such as the power button, were discussed. Then the two driving methods were explained to the subject. After this introduction, the subject was seated in the wheelchair and the user interface was connected to the wheelchair. The single switch scanning interface was explained to the subject, who then practiced using the interface with the motors turned off.

Once the subject was comfortable with the interface, the session entered a practice phase in which the subject first tried robotic assisted control and then standard manual control. The subject practiced both methods until he expressed an understanding of each control method; subjects usually spent about two minutes trying each method. All practice was done off of the test course, so that the subject was not able to learn anything that would assist him during the test phase.

The course (shown in Figure 1) was designed to include obstacles (several couches and chairs, a fire extinguisher mounted to the wall 30 cm (11.8 inches) above the ground, a trash can, and a table) and turns to the left and to the right. The course is 20 meters (65.7 feet) long. Three doors in the

	Manual				Robotic			
	First Run		Second Run		First run		Second run	
Number of clicks	90.2	(16.3)	77.1	(9.8)	25.6	(4.9)	22.0	(3.3)
Scanning time (sec)	93.6	(20.3)	81.1	(13.0)	30.9	(8.3)	25.2	(8.6)
Moving time (sec)	311.6	(36.4)	316.6	(36.2)	268.2	(21.5)	277.1	(28.4)
Total time (sec)	405.1	(42.1)	397.7	(43.7)	299.1	(18.4)	302.3	(32.5)

Table 1: Results of the indoor user tests: the number of clicks, amount of time spent scanning for commands, amount of time moving and total time to complete the course. The first number for each method is the mean and the number in parentheses is the standard deviation.

hallway on the course could be open or closed, determined by the office occupants.

The test phase consisted of four up-and-back traversals of the test course, alternating between the two control methods. Half of the subjects started with robotic assisted control and the other half started with standard manual control. Each up-and-back traversal consists of two parts: running the course from the couch area to the hallway and then the return trip. The turn in the middle of the course is not counted as part of the run, as turning completely around in the middle of the hallway is not a normal driving occurrence. The total session time for each subject was approximately 45 minutes.

There were four experimental performance measures collected by the computer that was running the user interface: (1) the number of clicks required to navigate the course, (2) the amount of time spent scanning to get to the necessary commands, (3) the amount of time spent moving or executing the given commands, and (4) the total amount of time spent on the course (scanning time plus moving time). The researcher only recorded the number of scrapes made by the chair. At the completion of the test, the user was asked to rank standard manual control and robotic assisted control on a scale from 1 (worst) to 10 (best).

Data for each experimental measure was analyzed using an ANOVA test. The differences between robotic control and manual control were highly significant with $p<.0001$ for all measures. On average, robotic control saved 60 clicks over manual control, which is a 71% improvement. Total time for robotic assisted control was 101 seconds shorter than manual control on average, which is a 25% improvement.

The only performance measure not collected on the computer was the count of the number of scrapes. A scrape was recorded when the chair brushed along a wall or piece of furniture. Bumps with the bumper were also counted as scrapes. No subject hit a wall or an obstacle with great force. The average number of scrapes per run under manual control is 0.25. The average number of scrapes under robotic control is 0.18. These numbers are not significantly different.

Finally, the subjects were asked to evaluate the two driving methods by giving a score from 1 (worst) to 10 (best). The average score for standard manual control was 3.5. The average score for robotic assisted control was 8.7. These scores are highly significant with $p<.0001$. No test subject preferred manual control over robotic control.

Subjects drove more efficiently and preferred to drive with robotic assisted control. Robotic control automatically adjusts for drift where manual control does not. When traveling down a long hallway under robotic control, a user can click on forward at the beginning of the corridor and does not need to do anything more until he wishes to stop or turn. Under manual control, the user must make many adjustments to compensate for drift.

The total time taken on a test run is a sum of the scanning time and the command execution time. Both scanning time and execution time improved from manual control to robotic control. As would be expected, if fewer clicks are issued, the scanning time required is shorter. Estimating that forward is clicked 50% of the time, left is clicked 25% of the time and right is clicked 25% of the time[2], with a scan time of one second and an estimated reaction time of one half second in our able-bodied subjects, each click would require an average of 1.25 seconds. As Table 1 shows, the scanning time is approximately 1.25 times the number of clicks.

Each user executed two trials for each control method. The differences between the two trials were significant for clicks ($p=.003$) and for time spent scanning ($p=.015$), indicating that the subjects were improving due to learning.[3] As the user became more comfortable with the system, he was able to judge more effectively when it was necessary to make adjustments to the current course.

A single subject ran the course 10 times in manual mode to determine how learning could affect the number of clicks and scanning time. The subject was this researcher; a naive user is not required to test for optimal performance. Over the 10 runs, the average number of clicks in a test run was 71.4 with a standard deviation of 9.5. Over the last 5 runs, the average number of clicks was 68 with a standard deviation of 4. Scanning time averages 73.7 seconds (standard deviation

---

[2] Empirically, backwards commands are issued very infrequently.

[3] There was no significant effect of learning on moving time and total time between trials; since the speed is held constant throughout the experiment, the user can not significantly reduce the amount of time required to travel the course between trials of the same control method.

12.0) over all 10 runs and 68 seconds (standard deviation 5.3) over the last 5 runs. Optimal performance for this course in manual mode will not approach the average performance in robotic mode.

5 COMPARABLE ABLE-BODIED TESTS

The NavChair system was tested in an indoor environment using voice control as the access method [Simpson and Levine 1997]. Six able bodied subjects navigated through three different scenarios (room traversal, door passage and wall following), four times with navigation assistance and then four times with no navigation assistance. For each scenario, it took longer to navigate using navigation assistance, primarily because the chair slows down as it gets closer to obstacles. However, no collisions occurred with navigation assistance, while there were occasional collisions with no navigation assistance. Test subjects preferred driving with navigation assistance.

The VAHM Project was also tested in an indoor environment using single switch scanning as the access method [Bourhis and Pino 1996]. Four able bodied users familiar with computers drove the wheelchair through a course simulating a kitchen and living room environment in manual mode and in an assisted mode which provided obstacle avoidance. Assisted mode resulted in a 13% improvement in the number of actions required on the interface screen (which we called the number of clicks required). Execution time also improved an average of 7.7%. The tests were executed at three scanning rates: 0.8, 2.5 and 4.5 seconds. There was a 2.5% improvement in execution time for the 0.8 second rate,[4] a 7.7% improvement for the 2.5 second rate, and a 13% improvement for the 4.5 second rate. One would expect to see a more dramatic improvement in total execution time for longer scanning times since fewer clicks result in a greater time savings.

Indoor user tests of Tin Man II used a joystick and buttons for the access method [Miller and Slack 1995]. The test course was 50 meters long and included a hallway, a doorway and two rooms. Subjects were told to attempt to minimize their joystick movements in both the manual and assisted tests. Manual mode required 50% more joystick moves than the obstacle avoidance mode. The time required to traverse the course was less than 10% longer in assisted mode than in manual mode.

6 CONCLUSIONS

User tests are required for assistive navigation systems. These systems are designed for people to ride; it is important that a great deal of testing occurs from the early design stages to the end. Able-bodied test subjects should be utilized for all testing until the system is demonstrated to be safe, reliable and useful. Only after a long period of testing on able-bodied subjects should the testing proceed to the target population.

When designing assistive technology, providers should be asked for comments on the system from the initial design stages through final testing of the product. The inclusion of providers will result in designs that better reflect the needs and desires of the target population. Providers can also facilitate safe testing of the target population.

Performance metrics for assistive navigation systems are necessary to demonstrate the usefulness and usability of the system. Assistive navigation systems are medical devices; as such, they should be held to the same strict testing guidelines mandated by the FDA, even in the initial design phases.

ACKNOWLEDGMENTS

The Wheelesley user tests were performed while the author was a graduate student in the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. This research was funded by the Office of Naval Research under contract number N00014-95-1-0600 and the National Science Foundation under grant number CDA-9505200.

REFERENCES

[Bourhis and Pino 1996] G. Bourhis and P. Pino. "Mobile robotics and mobility assistance for people with motor impairments: rational justification for the VAHM project." *IEEE Transactions on Rehabilitation Engineering*, 4(1): 7 – 12, March 1996.

[Dallaway et al. 1995] John L. Dallaway, Robin D. Jackson, and Paul H. A. Timmers. "Rehabilitation robotics in Europe." *IEEE Transactions on Rehabilitation Engineering*, 3(1):35 – 45, March 1995.

[Kazi et al. 1998] Zunaid Kazi, Shoupu Chen, Matthew Beitler, Daniel Chester, and Richard Foulds. "Speech and gesture mediated intelligent teleoperation." In *Lecture Notes in Artificial Intelligence: Assistive Technology and Artificial Intelligence*, edited by Vibhu Mittal, Holly A. Yanco, John Aronis, and Richard C. Simpson, Springer-Verlag, pp. 194 – 210, 1998.

[Lacey 1999] Gerard Lacey. *Personal Adaptive Mobility Aid for the Frail Visually Impaired.* Ph.D. Thesis, University of Dublin, Trinity College, April 1999.

[Miller and Slack 1995] David P. Miller and Marc G. Slack. "Design and testing of a low-cost robotic wheelchair prototype." *Autonomous Robots*, 2:77 – 88, 1995.

[4] This is the closest to our 1 second scanning rate.

[Simpson and Levine 1997] Richard C. Simpson and Simon P. Levine. "Development and evaluation of voice control for a smart wheelchair." In *Proceedings of the Rehabilitation Engineering Society of North America 1997 Annual Conference*, pp. 417 – 419, RESNA Press, 1997.

[Wagner et al. 1999] J.J. Wagner, H.F.M Van der Loos, N. Smaby, K. Chang, and C. Burgar. "ProVAR assistive robot interface." In *Proceedings of ICORR '99: International Conference on Rehabilitation Robotics*, pp. 250 – 254, 1999.

[Yanco and Gips 1998] Holly A. Yanco and James Gips. "Driver performance using single switch scanning with a powered wheelchair: robotic assisted control versus traditional control." In *RESNA-98*, pp 298 – 300, 1998.

[Yanco 2000] Holly A. Yanco. *Shared User-Computer Control of a Robotic Wheelchair System.* Ph.D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 2000.

Measuring Classifier Intelligence

Jim DeLeo
Department of Clinical Research Informatics
NIH Clinical Center
National Institutes of Health
Bethesda, Maryland 20892
jdeleo@nih.gov

ABSTRACT

Classifiers are seen here as systems in which input feature values are used with fitted or learned functions that produce output values which are interpreted as probabilities or fuzzy degrees of class membership, or in which output values are used with cut-off decision rules to choose bivalent class membership. Two complementary measurements for evaluating training, validation, testing, and deployment phase performances in human, mechanical, and computerized classifiers are proposed here. These measurements are derived from samples of classifier output values paired with their corresponding known probabilistic, fuzzy, or bivalent classification values. The first measurement is the area under the ROC plot. The second is the separation index newly introduced here. Both of these measurements are easy to understand and to compute. It is proposed that they be considered standard metrics for evaluating and comparing classifier intelligence.

Keywords: *classifiers, intelligence, performance metrics, intelligence metrics, area under the ROC plot, separation index, knowledge discovery from data, ensembles*

1. Introduction

The task of a human, mechanical, or computerized classifier is to use a set of values, x's, for certain particular attributes to classify an entity or event into one or more categories or classes. Classification may be bivalent, where the classifier output, y, is either negative (y=0) or positive (y=1), probabilistic where the output is a probability ($0 \leq y \leq 1$) that the entity or event is associated with the bivalent positive class, or a fuzzy degree of membership ($0<y<1$) reflecting partial degree of membership in the positive class. As classification tasks become increasingly non-trivial with many attributes and highly complex nonlinear and discontinuous relationships among attribute values and classification outcome values, it may be said, in the spirit of classical artificial intelligence, that classifiers that perform well are demonstrating intelligence. Metrics are needed to measure this intelligence in order to describe and compare classifier performances. Here, two such metrics that complement each other are proposed. The first is the fairly well known area under the ROC plot. The second is a new index called the "separation index." Both metrics may be employed for bivalent, probabilistic, and fuzzy classifier outcomes. They have immediate use with present day classifiers and they have potential future use with anticipated large ensembles of autonomous intelligent classifier agents engaged in data mining for knowledge discovery purposes by means of perpetual dynamic exploration of large and expanding data bases.

2. Classifiers and Intelligent Metrics

A basic classifier is illustrated in Figure 1.

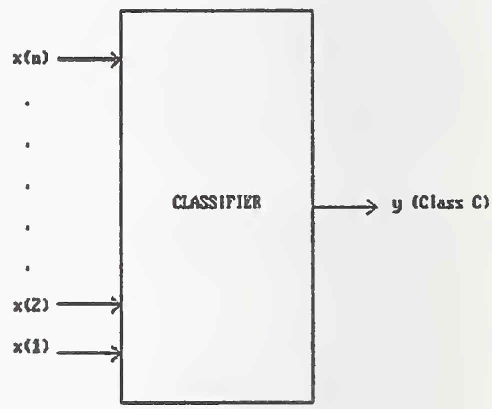


Figure 1. Illustration of a basic classifier that maps a fixed finite set of input parameter values, x's, into an output parameter value, y, where y is a measure of bivalent, probabilistic, or fuzzy classification of an entity or event with respect to some specific class, C.

Its purpose is to map a fixed finite set of input parameter values, x's, associated with an individual entity or event, E, into an output parameter value, y, where y is a measure of association of E with respect to some specific class, C associated with the output node that produces y. Basic classifiers may be designed so that y-values reflect bivalent, probabilistic, or fuzzy classification associations. Classifiers may have more than one output node where each node is associated with a different outcome class. The intelligence metrics described here may be applied to each separate output node. Experience suggests, however, that in polyvalent or n-class classifier applications it may be wiser to construct n basic classifiers, each having a single output node, instead of constructing one classifier with n output nodes. This is because the performance of internal mathematical "features" is reduced when these features must be "shared" in the computation of multiple outputs.

There are many mathematical methods for developing classifiers. These range from simple function fitting techniques to highly sophisticated statistical and neural network ensemble modeling. The intelligence performance metrics described here, the area under the ROC plot and the separation index, are applicable to all underlying mathematical models used in classifiers. Classifiers undergo training, learning, or fitting - terms generally used interchangeably. Classifiers also undergo validation, testing, and deployment phases. The intelligence metrics described here are intended for use in all of these phases.

3. Classifier Output Interpretation

A classifier should be designed from the outset to perform bivalent, probabilistic, or fuzzy classifications, and used in the same way throughout training, validation, testing, and deployment phases. The fundamental epistemological and mathematical differences among these three classifier types must be clearly understood at the outset of designing a classifier. These differences are based on the understanding and interpretation of the output parameter, y, and this interpretation is based on the meaning given to the class assignment data used in developing and using the classifier.

If membership data is bivalent, meaning that entities or events are perceived as belonging discretely to one or the other of the positive or negative bivalent classes, then y-values must also be interpreted as bivalent, negative or positive, usually expressed as 0 and 1 respectively. Classifiers trained with bivalent class data will often produce continuous output values for y on these intervals during all classifier phases. When this is the case, threshold decision rules are needed to force bivalent classification. If membership data is probabilistic, meaning that entities or events are perceived as belonging probabilistically to the positive pole of the bivalent classes, then output y-values may be interpreted directly as probabilities of positive class membership. For example, y= .8 could mean there is a .8 probability that the patient is a member of the bivalent set “bivalent diabetics.” If membership data is fuzzy, meaning that individual entities or events are perceived to be partly in the positive class and partly in the negative class, then output values should also be interpreted as fuzzy membership values [1]. In this case, y=.8 could mean that the patient has a .8 degree of membership in the fuzzy set "fuzzy diabetics."

Again, once a classifier system is designed to be bivalent, probabilistic, or fuzzy, it should be considered that way during training, validation, testing, and deployment phases. The interpretation of the classifier output must remain consistent throughout all of these phases.

4. The Area Under the ROC Plot

The first proposed metric of classifier intelligence is the area under the ROC plot. It is derived from ROC methodology which has origins in signal detection theory [2,3]. ROC methodology addresses forced choice bivalent classifications [4-7]. The "receiver" is a human, mechanical, or computerized agent performing the bivalent classification. "Operating characteristic" refers to the performance of the receiver. The central feature of ROC methodology is the ROC plot constructed from bivalent frequency distribution data specified as independent variable values, y, paired with dependent known classification values y_k where y_k=0 means full membership in the negative class and y_k=1 means full membership in the positive class. A basic bivalent classifier developed, for example, with neural network methodology, will have an output variable, y, with continuous values on the 0-1 interval. For purposes of applying ROC methodology, this output variable y is the independent variable which when coupled with the known classification values, y_k=0 and y_k=1 provides the data with which to compute a ROC plot. Figure 2 shows simulated bivalent frequency distributions of output values from a neural network classifier at the completion of a successful training operation. In this figure, the abscissa variable is y, the continuous neural network classifier output variable. The ordinate is the frequency at which various y values occur in both the negative class where y_k=0 (grey bars), and in the positive class where y_k=1 (black bars). It is apparent from these contrasting distributions that there are approximately the same total number of negative cases as there are positive cases. This means that the prevalence or incidence of positive events is approximately .5 in the training data. Special consideration needs to be given when there is a mismatch of prevalence in data used in training and deployment. Additional special consideration needs to be given to misclassifications cost differences, meaning differences in false positive and false negative costs. Prevalence and misclassification cost issues are very important. They have been addressed elsewhere, however more work is needed [8-12].

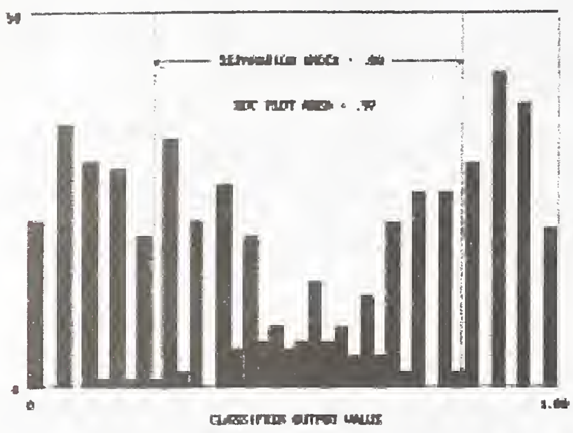


Figure 2. Bivalent frequency distributions of output values, y, from a classifier at the completion of a successful classifier neural network training operation.

The ROC plot is a plot of sensitivity versus 1-specificity as the independent variable traverses its full range. In the case of the classifier, the independent variable is y, the output of the classifier, and it ranges continuously from 0 to 1. Specificity is computed as the normalized (scaled to 1) integral of the negative distribution. Sensitivity is computed as the normalized integral of the positive distribution subtracted from 1. The ROC plot computed from the data summarized in the frequency distributions in Figure 2 is the ROC plot with an area of .97 hugging the upper left corner of the grid in Figure 3. It is displayed in this figure with 4 other ROC plots that were constructed at earlier stages in the training process. These ROC plots are empirical ROC plots meaning that they are directly computed from the y, y_k paired data. Since sensitivity and specificity are computed independently, ROC plots are, in a sense, prevalence independent.

The area under the ROC plot is readily computed by numerical integration. This area is a statistic. It is the probability that a randomly drawn event associated with the positive class will have a higher value for the independent variable, y, than a randomly drawn event associated with the negative class. In statistics it is computed as the Mann-Whitney version of the Wilcoxan statistic. The ROC plot area ranges from 0, which indicates full separation with positive cases having lower y values than the negative cases, through .5 which indicates the poorest performance (no separation of bivalent classes), to 1 which indicates the full separation of bivalent classes with positive cases having the higher y values. Classifiers with high ROC plot area values may be further differentiated in performance by means of the separation index.

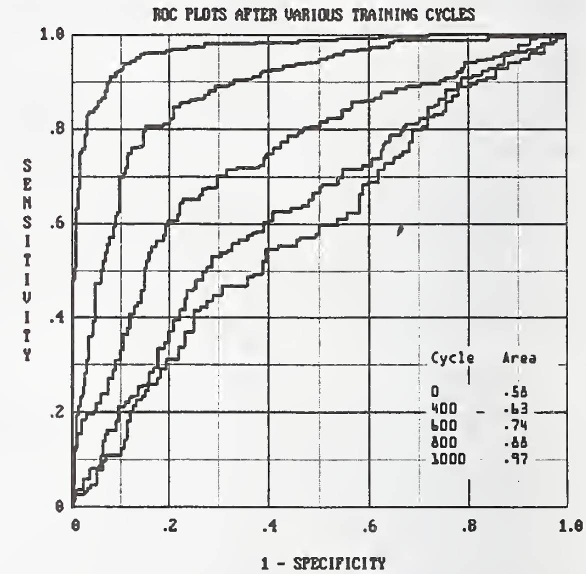


Figure 3. ROC plots from simulated neural network classifier output data after 0, 400, 600, 800, and 1000 training cycles.

5. The Separation Index

The second measure of classifier intelligence is the separation index introduced here. The separation index is a measure of the difference between the median y-values of the positive and negative frequency distributions. It is computed by first determining the median y-value for all negative cases, n_{med}, and the median y-value for all positive cases, p_{med}. Subtracting n_{med} from p_{med} yields a value on the -1 to +1 interval. To map this value onto the 0 to 1 interval, 1 is added and the result is divided by 2. The formula for the separation index (SI) is as follows:

$$SI = (p_{med} - n_{med} + 1) / 2 \qquad (1)$$

6. Index Complementarity

The ROC plot area and the separation index both directly measure class separation whereas other measurements used in developing classifiers generally measure the fitness of the data to the underlying function. For example, the root mean squared (RMS) error measures the square root of the sum of the squares of the differences between known, y_k, and fitted, y, outcome class values. This is clearly not a direct measure of separation. Since the task of a classifier is separation and since performing this task well requires intelligence, the ROC plot area and the separation index may be justifiably thought of as measures of intelligence since they both directly measure separation. Furthermore these indices complement one another. For example, if a ROC plot area of 1.00 indicating full separation is obtained, the separation index may be used to further differentiate training states or to compare classifiers. If poor ROC plot areas are obtained, the separation index could again indicate better or worse separation in comparing different training states or different classifiers.

7. Indices for All Classifier Phases

The ROC plot area and the separation index may be computed for the training, validation, testing, and deployment phases of classifiers for purposes of classifier evaluation and inter-classifier comparisons. Plotting values of these indices after each training cycle provides a graphical representation of the rate of intelligence development ("learning curves") during training as well as other characteristics of training, such as stalling, nonmonotonicities, and reversals. Figure 4 illustrates training plots for 1000 training cycles in the simulated experiment referred to earlier, and Table 1 contains a partial tabular listing of the data plotted in Figure 4.

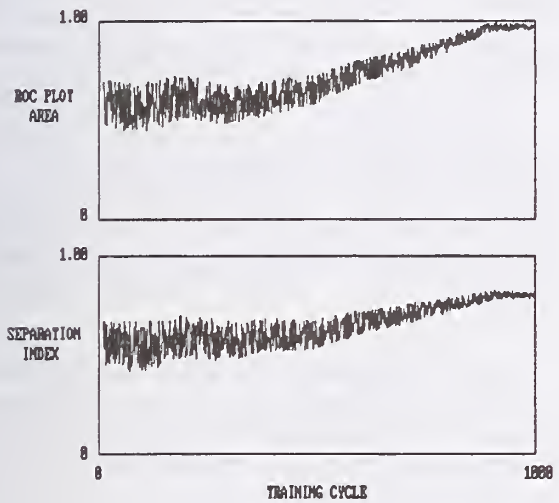


Figure 4. ROC plot area and separation index values in a simulated 1000 cycle neural network classifier training operation. (These might be thought of as "learning curves.")

What is being simulated here is the training of a neural network in which individual training cases are selected by bootstrap sampling [13-16]. This random sampling with replacement strategy is what gives rise to early higher variances tapering off to later lower variances for both indices over the training cycles. Experience suggests that this kind of sampling provides a weight jogging effect which aids in avoiding local minima entrapment. Also of note in this simulation is the observation that index values idealistically approach an asymptotic maximum. Perhaps other new metrics for intelligence could be devised for training plot features such as variance tapering and asymptotic convergence.

If validation is pursued concurrently with training, perhaps use of the area under the ROC plot and the separation index as new intelligence metrics will yield new ideas about training termination. Perhaps the intelligence metric values derived from testing data evaluated after training and validation will be considered the appropriate values to use for comparing classifiers. Monitoring intelligence metric values periodically during deployment operations would be a good way to assure that the trained classifier is holding up and that environmental data sources are not drifting too far from the original data populations associated with training, validation, and testing data sources

TRAINING CYCLE	ROC PLOT AREA	SEPARATION INDEX
0	.58	.57
100	.56	.54
200	.62	.59
300	.57	.57
400	.62	.60
500	.65	.60
600	.74	.66
700	.80	.70
800	.88	.75
900	.97	.80
1000	.97	.80

Table 1. ROC plot area and separation index values in a simulated 1000 cycle neural network classifier training operation.

8. Fuzzy and Probabilistic Membership

ROC methodology can be easily extended to include fuzzy and probabilistic classifications [17,18]. This is done by simply considering every entity or event as having relationship to both the negative and the positive bivalent poles of the class associated with the dependent variable, y_k. Let the membership association value be y_k for the positive class, and $1-y_k$ for the negative class. Thus, a fuzzy or probabilistic membership value of $y_k=.82$ corresponds with a .82 association value in the positive class and a .18 association value in the negative class. This simple generalization subsumes classical ROC methodology, because $y_k=1.00$ corresponds to an association value of 1.00 in the positive class and 0 in the negative class, and $y_k=0$ corresponds to an association value of 0 in the positive class and 1.00 in the negative class. After positive and negative class association values are determined, sensitivity and specificity values are computed from the resulting bivalent frequency distributions and the ROC plot is computed from these sensitivity and specificity values as before. The area under the ROC plot is computed by numerical integration as before or by a weighted Mann-Whitney version of the Wilcoxan statistic with tied data [18]. Likewise, the separation index is derived from the resulting bivalent frequency distributions as before.

9. Discussion

Using metrics such as those proposed here to measure classifier intelligence for evaluating and comparing classifier systems will become increasingly important in environments where large cadres of automated intelligent agents will be used in knowledge discovery from data efforts by continuously data mining large and expanding data bases. Efficient algorithms for computing intelligence metrics will be needed in these environments.

Intelligence measurement could perhaps be only one kind of measurement appropriate for evaluating intelligent systems such as classifiers. Design simplicity, computational ease, computational speed, and the capability to map knowledge produced with machine intelligence to human understandable knowledge are other important features for which metrics could be developed. Developing such measures could be an important step foreword in the progression of machine intelligence. Perhaps this step will help expand human knowledge and understanding in a more general way. By understanding intelligence and related characteristics in machines, humans may come to better understand these characteristics in humans.

10. Conclusions

Two complementary metrics, the area under the ROC plot and the separation index, have been shown to be effective measures of intelligence in all phases of classifier systems that produce bivalent, fuzzy, or probabilistic classifications. It is proposed that these metrics be standardized as measures of classifier intelligence for purposes of evaluation and comparison.
The need for fast algorithms for assessing intelligence has been suggested as well as the need for measuring other attributes of classifiers and other intelligent agents, specifically attributes related to parsimony, and human knowledge derivation. The need for more work on prevalence and misclassification cost issues in classifiers has also been mentioned. It has been suggested that understanding anthropomorphized characteristics in machines may promote understanding of related characteristics in humans.

References

[1] Kosko, B., Fuzzy Thinking. New York: Hyperion, 1993.

[2] Green, D.M., Swets J.A., Signal Detection Theory and Psychophysics. New York: John Wiley & Sons, Inc., 1966.

[3] Lusted, L.B., "Signal Detectability and Medical Decision-making," Science, vol. 171, pp. 1217-1219, 1971.

[4] DeLeo, J.M. "Receiver Operating Characteristic Laboratory (ROCLAB); Software for Developing Decision Strategies That Account for Uncertainty," in Proceedings of the Second International Symposium on Uncertainty Modeling and Analysis, IEEE Computer Society Press, 1993, pp. 318-325..

[5] DeLeo, J.M. "The Receiver Operating Characteristic Function as a Tool for Uncertainty Management in Artificial Neural Network Decision-Making," in Proceedings of the Second International Symposium on Uncertainty Modeling and Analysis, IEEE Computer Society Press, 1993, pp. 141-144.

[6] Lusted, L.B., "ROC Recollected," Medical Decision Making, vol. 4, pp. 131-135, 1984..

[7] Zweig, M, Campbell, G., "Receiver Operating Characteristic (ROC) Curves: a Fundamental Tool in Clinical Medicine," Clinical Chemistry, vol. 39, pp. 561-577, 1993.

[8] DeLeo, J., Dayhoff, J., "Medical Applications of Neural Networks: Measures of Certainty and Statistical Tradeoffs," in Proceedings of the International Joint Conference on Neural Networks (IJCNN'01) IEEE Press, 2001, pp. 3009-3014.

[9] DeLeo, J., Rosenfeld, S., "Essential Roles for Receiver Operating Characteristic (ROC) Methodology in Classifier Neural Network Operations," in Proceedings of the International Joint Conference on Neural Networks (IJCNN'01) IEEE Press, 2001, pp. 2730-2731.

[10] DeLeo, J.M., Rosenfeld, S.J., "Important Statistical Considerations in Classifier Systems," in Proceedings of the Fourteenth IEEE Symposium on Computer-Based Medical Systems, IEEE Computer Society, 2001, pp. 285-293.

[11] Dayhoff, J.E., DeLeo, J.M., "Artificial Neural Networks: Opening the Black Box," Cancer 2001 Apr 15; vol. 91 Suppl. 8, pp. 1615-1635, 2001.

[12] Remaley, A.T., Sampson, M.L., DeLeo, J.M., Remaley, N.A., Farsi, B.D., Zweig, M.H., "Prevalence-Value-Accuracy Plots: a New Method for Comparing Diagnostic tests based on Misclassification Costs," Clinical Chemistry, vol. 45(7), 1999, pp. 934-941.

[13] Efron, B., Tibshirani, R.J., An Introduction to the Bootstrap. New York: Chapman and Hall, 1993.

[14] Efron, B., The Jackknife, the Bootstrap and Other Resampling Plans. Montpelier, Vermont: Capital City Press, 1994.

[15] Hall, P., The Bootstrap and Edgeworth Expansion. New York: Springer-Verlag, 1992.

[16] Campbell, G., DeLeo, J.M., ``Bootstrapping ROC Plots," in Computing Science and Statistics: Proceedings of the Twenty-Seventh Symposium on the Interface, 1995, pp. 420-424.

[17] DeLeo, J.M., "The Fuzzy Receiver Operating Characteristic Function and Medical Decisions with Uncertainty," in Proceedings of the. First International Symposium on Uncertainty Modeling and Analysis, IEEE Computer Society Press, 1990, pp. 694-699.

[18] Campbell, G., DeLeo, J.M., "Fundamentals of Fuzzy Receiver Operating Characteristic (ROC) Functions," in Computing Science and Statistics: Proceedings of the Twenty-First Symposium on the Interface, 1989, pp. 543-548.

Problems of Performance Measurement in Locally-Organized Systems

Vadim L. Stefanuk
Institute for Information Transmission Problems
Russian Academy of Science
Bolshoy Karetny per. 19,
Moscow, 101447, Russia

ABSTRACT

It is shown that many of the modern intelligent systems belong to a wide class of distributed systems. The external behavior of such a system is governed with criteria which induce only partial ordering among various systems. This partial order does not allow building an analytical metrics in the space of such systems, the fact making systems largely indistinguishable. This explains the existence of many versions of AI systems made for the same purpose, like Expert Systems shells, which are frequently differentiated only through their secondary properties. The situation may be compared with that of the use of Pareto sets in theory of games where all different solutions belonging to a Pareto set are considered to be intrinsically similar.

KEYWORDS: *distributed systems, performance metrics*

1. INTRODUCTION

Practical considerations do require introduction of some metrics to measure system performance. If a scalar value is used to estimate the performance it is quite natural to define a metrics based on this scalar. In some cases there are some serious grounds for this. The information throughput [1], an average income obtained in the system for Stock Exchange trend forecast may be considered as examples of the scalar valued system.

However when the performance is measured by a vector value it obviously brings some additional problems. In the simplest case a weighted vector is used reducing the problem to the scalar type formulation. However in many cases it is impossible to find weights uniformly suitable for the whole performance space like the distance measure in an Euclidean space. It is for this reason some other "optimality" considerations are used like Pareto sets in Game Theory [2] and in many other applications (in particular, in Mathematical Economics.)

2. DISTRIBUTED SYSTEMS

Another wide area where the search for a performance metrics is doomed to failure, and which will be discussed in this presentation, is the area of distributed systems, referred here to as the locally-organized systems (LOS) [3, 4]. We use the latter term in order to emphasize the fact that in such systems a unique common measure defining the whole system performance does not exist. Instead, one has a collection of local criteria for the system components [5], which are used concurrently and individually and hence can not be reduced to a unique scalar or vector value.

The exact meaning of notions, used in our description of locally organized systems depends on the subject domain. However it is possible to demonstrate some classes of models for which the situation is known in advance:

- The subsystems are deterministic or probabilistic finite automata with the deterministic or probabilistic interaction (collective behavior, collective behavior of automata, automata games)
- Subsystems are automata with the continuous sets of actions with the interaction of deterministic type (multiple access communication systems, sociological and economics models).
- Subsystems are finite automata with a fuzzy interaction (Expert Systems).
- Subsystems are the enterprises or individuals who are involved in a complex monetary and commodity circulation.
- Inhomogeneous technical systems, where people are also involved (man-machine systems).
- Subsystems are interacting programming modules (interactive high level languages, Artificial Intelligence systems).

Anyway, probably due to some practical considerations, many distributed systems of this kind are treated as centralized ones allowing some metrics for the whole system

This research was partially supported by grant from the program "Mathematical modeling, Intellectual computer systems and non-linear mechanical systems control" of Russian Academy of Sciences, project # 2.24, and by grant from Russian Fund for Basic Research, project #02-01-00955.

performance. Thus, in the majority of multi-agent systems [5] there exists a central point collecting all the information from agents and assigning tasks to the individual agents. It is important to note that such a centralization is out of question for LOS.

The proposed presentation will be organized in the following way. First, we will introduce a formal definition for LOS. Then we will list the reasons why LOS becomes a necessity in many applications, later describing the methods typically used for analyses and study of LOS.

We still are able to speak about "goals" for LOS. However the goal of the locally organized system (LOS) is the provision of a normal functioning of its subsystems, instead of reaching a certain system goal. That is why the criteria for the choice of an adequate system version may include such considerations as *expediency*, *survivability*, *openness*, *elasticity* and etc., which have been studied in many applications and in many subject domains [6] and which will be briefly reviewed in our presentation. Any of the above criteria may take only one of two values Yes/No or 1/0. Obviously, the induced partial order gives only limited possibility to compare systems.

The mentioned criteria are subject to changes from one group of problems to another still having the property that they do not allow to compare systems by introducing any reasonable metrics to supplement the above step-wise partial order. A number of examples will be listed - from collective behavior to manipulator control and interacting programming modules of an Artificial Intelligence system. This list and other considerations will demonstrate that the fraction of systems belonging to the class of LOS will probably only grow with time.

Finally some mathematical models of an intelligent warehouse [7] are used to illustrate the whole approach in a step by step manner.

3. CONCLUSION

As many of the modern intelligent systems are organized in a local way, being collectives of interacting components each having its own goal and behavior, our results show that it is not simple to find a suitable metrics for measuring system performance to be used for making a comparison among intelligent systems with respect to each other. Probably this explains the existence of many versions of AI systems made for the same purpose, like Expert Systems shells, which are frequently differentiated only through their secondary properties.

Our analyses is formal and strict. However it is supplemented with a practical example of an intelligent warehouse which admit different type of organization of storage place.

4. REFERENCES

[1] Fano R.M., Transmission of information. A statistical theory of communication., MIT Press, N.Y., 1961.

[2] Luce R.D. & Raiffa H. Games and Decisions. Introduction and critical survey, Harvard University, N.Y., 1957.

[3] Stefanuk V.L. Collective Behavior of Automata and the Problems of Stable Local Control of a Large Scale System, Second International Joint Conference on Artificial Intelligence (IJCAI), London, pp. 51-56, 1971.

[4] Stefanuk V.L. "General Approach to Locally-Organized Systems", Proceeding of Advanced Summer Institute 1999 (ASI'99) of the Network of Excellence in Intelligent Control and Integrated Manufacturing Systems, Leuven, Belgium. pp. 39-45, 1999.

[5] Proceedings of IJCAI-01, "Multi-agent systems", pp. 1017-1224.

[6] Stefanuk V.L., Stability of Local Control in a System with Contralinear Interaction of Subsystems, Proceedings of European Control Conference (ECC'93), vol. 1, Groningen, pp. 117- 119, 1993

[7] Stefanuk V.L., Algorithms to Optimize Retrieval in a Distributed Memory System, Proceedings of RAN: Theory and Control Systems, #4, 1999. - pp. 93-99.

PART I
RESEARCH PAPERS

IM2 - Performance of Multiple Agents

A CONTROL SCHEME FOR MEASURE OF PERFORMANCE AND EFFICIENCY OF TACTICAL COOPERATIVE ROBOTS

Amir Shirkhodaie
Intelligent Tactical Mobility Robotic Research Laboratory
Department of Mechanical & Manufacturing Engineering
Tennessee State University
Ashirkhodaie@TNState.edu

ABSTRACT

We have discussed technical issues regarding supervised world perception modeling and task planning of cooperative mobile robots performing tactical tasks in unstructured environment. We have introduced a hierarchy Supervisory Controller for robust cooperative task deployment of heterogeneous semi-autonomous robotic vehicles. Primarily, we have described functional and modular architecture of the Supervisory Controller and presented our strategies for separation of supervisory functions according to their level abstraction, complexity, precedence, and intelligence. Furthermore, we have discussed control schemes for measure of performance and measure of efficiency of our cooperative robots tested under different task situations. Results indicate the hybrid Supervisory controller performs satisfactorily in most simulated cases.

Keywords: Cooperative Robots, Supervisory Control, Measure of Performance, and Measure of Efficiency.

1. INTRODUCTION

Cooperative robots are critically important for solving a number of real-world problems. Their applications span from civilian search and rescue missions to military battlefield reconnaissance and surveillance covert operations to deep space scientific research explorations. In very near future, potential commercial applications of cooperative robots are foreseen to be ubiquitous. The cooperative robots while benefit from inherent parallelism, their robust control is both multi-folded and challenging to achieve and sustain reliably. Many different techniques and approaches for mobile robots control have been developed. The proposed control techniques in literature can be, categorically, classified as either deliberative, reactive, or hybrid in nature.

The control schemes that tend to be more deliberative require relative more knowledge about the world. They use this knowledge to predict the outcome of their actions, an ability that enables them to optimize their performance relative to their model of the world. Deliberative reasoning about task planning of cooperative robots requires strong assumptions about this world model. Primary knowledge upon which the reasoning is based on, should be consistent, reliable, and certain. In a dynamic world, where objects may have arbitrarily moves (i.e., in a battlefield or a crowded hallway), it is potentially risky to rely on information that no longer be valid. Instead, world representational models are generally constructed at run-time using a combination of past knowledge gained about the environment and exteroceptive sensory data.

At the other end of the spectrum, the reactive control systems attempt to tightly couple perception and action in order to achieve faster robot response in dynamic and unstructured worlds while minimizing computational overhead. With a purely reactive control system, it is rather difficult to achieve planned deliberated tasks consistently. This is mainly due to variability in world uncertainty and lack of robot's knowledge and ability in resolving conflicts between competitive world perceptions in a given situation. In most cases, a wrongly selected world perception may cause the robot an unrecoverable deadlock situation or failure.

The temporal inconsistency and stability of the environment and the robot's immediate sensing inadequacy for a task are typically coupled. Difficulty in proper localization of a robot and the way that the robot perceives its surrounding world also yield possible situations that typically grounds erratic conflicts in decision making process of the robot. To demonstration this notion, consider figure 1 that illustrates spatial configuration of three cooperative robots. Should each robot presume the other two robots as obstacles or as its teammates approaching it unintentionally? One answer is it would be dependent on nature of the task and how the navigational modes of the robots are defined. If the robots had reactive behavior, they would probably try to find a way out of the crowd or avoid the deadlock situation. If the robots had deliberative behavior, each robot might refer to its chronicle memory and try to reason why the other robots are there in the first place before it makes any decision. Situation like this example can occur frequently during task execution of cooperative robots within a limited work environment. Hence, it is responsibility of an intelligent supervisory system to deal with such situations in a manner that causes least perplexity to plan execution of robots.

An overview of common conceptions of the behavior-based approaches is given by Mataric [1]. Brooks [2] describes four key concepts that lead to behavior-based robotic: situatedness, embodiment, intelligence, and emergence. The design of behavior-based systems is often referred to as a "bottom up" process, but this offers not so much to determination of the structure of the system as to a basis in physical sensing and

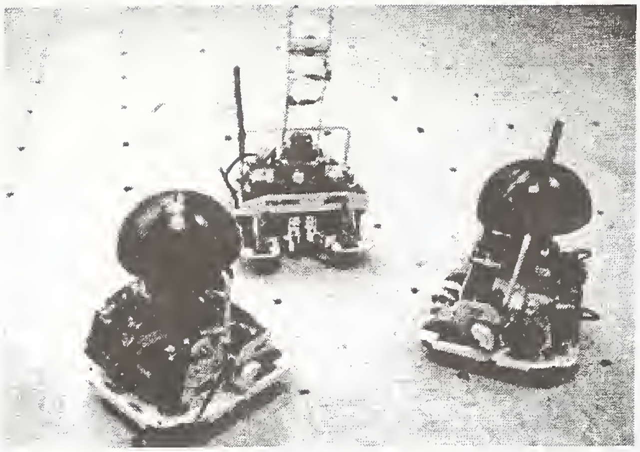


Figure 1. Three Semi-autonomous Cooperative Robots in a Tactical Formation.

action, and incremental development of sophistication from simple to complex. Namely, there are structured in terms of observable activity that they produce, rather than traditional functional decompositions. The activity producing components, behaviors, compete for actuator resources as well as share perceptions of the world rather than any centralized representation.

Recently, hybrid deliberative/reactive robotic architectures have emerged that rely more heavily on explicit world representations and tend to combine many aspects of traditional AI symbolic methods with situated-based reasoning. They operate based on abstract representation of the world in light of providing faster response, better robustness, and more tractable than deliberative and reactive systems. Hybrid architectures permit reconfiguration of reactive control systems based on available world knowledge through their ability to reason over the underlying behavioral components. However, building of such hierarchy systems requires compromise and full utilization of reactive and deliberative systems to maintain the desirable system performance and efficiency.

An overview of approaches and issues in cooperative robotics has been also reported in [3,4,5]. Parker [6] has demonstrated multi-robot target observation using the ALLIANCE Architecture, where action selection consists of inhibition (through motivation behaviors). As opposed to her architecture, Pirajanian and Mataric [7] have developed an approach to multi-robot coordination in the context of cooperative target acquisition. Their approach is based on multiple objective behavior coordination extended to multiple cooperative robots. They have demonstrated a mechanism for distributed command fusion across a group of robots to pursue multiple goals in parallel. This technique enables individual robot to select actions that no only benefit itself but also benefit the group as a whole. A significant amount of work in this area is being conducted by researchers at NIST. Their hierarchical control architecture is shown to have capability in controlling several unmanned mobile robotic vehicles in unstructured environment using a hierarchy platoon level control scheme where the robots follow a designated leader while maintaining a fair distance apart [14].

Nonetheless, intelligent control of multi-agent robots is both complex and challenging. The complexity of the task is contributed to a number of compounding factors including: multi-agent task decomposition, task distribution, resource allocation, sensory world perception modeling and data sharing, pattern recognition and reasoning, skill learning and adaptation, communication networking, man-machine interaction, and others. For intelligent strategic task planning, execution, and monitoring of cooperative robots, one should be concerned with many of above technical challenges.

In this paper, we will present a hybrid hierarchical deliberative/reactive robotic architecture called, *Supervisor Mobility Controller* - in short *"Supervisor"* for controlling a team of cooperative robots. By combining reactive and deliberative navigational schemes, we have created a set of group navigational techniques assisting task deployment of the robots. The Supervisor has been tested for localization and dynamic cooperative task planning of robots. Performance and efficiency of the system is measured on a physical robotic system consisting of five cooperative robots.

The proposed Supervisor control system has been developed under FMCell software [18]. FMCell provides tools for world perception construction and sensors modeling in 3D virtual simulation environment. Other features of the software include: high-level object-oriented environment with embedded robot behavioral modeling tools, fast image processing tools, AI-based inference engines for knowledge processing and reasoning, and soft computing developmental tools such as neural networks, fuzzy logics, and genetic algorithms for modeling, simulation and validation of control strategies.

2. SUPERVISOR ARCHITECTURE

The Supervisor was originally designed for control of semi-autonomous robots operating under one central control unit. The modular software implementation of the Supervisor is presented in Figure 2. Supervisor has a hierarchical architecture and designed to handle hybrid reactive/deliberative task deployment of the cooperative robots. Detailed description of different functions of this Supervisor is behind the scope of this paper and can be found elsewhere [8,9]. The hierarchy architecture consists of sensing, planning and acting components. The system allows direct interaction of the human operator at different levels of abstract task planning, execution, and monitoring with minimum restriction. An exclusive language allows for mission plans of the virtual robots with concise details. The supervisor can be used for control of both simulated and physical cooperative robots.. The same syntax as used for programming of physical robots is used for programming of the simulated robots. This feature significantly reduces development and implementation times and makes it submissive to both software and hardware requirements.

There are many good reasons for working with simulated robots: "learning often requires experimenting with behaviors that may occasionally produce unacceptable results," [10]. Making mistakes on real physical robots can be sometimes quite costly and dangerous. Nolfi et. Al. [17] suggests that simulated robots be used in developing control systems for real robots when certain special conditions are taken into account. However, one should not expect control systems that are evolved in a simulated environment to behave exactly the same in a real environment. Environment has many degrees of uncertainties, in particular in case of mobile robots where skidding and slippage are inevitable. To reduce this ambiguous problem, one can literally add certain degree of white noise to parameters having anticipated uncertainties. For instance, in dead-reckoning computation of robots, one may add random noise to offset encoders and compass readings of the robots and mimic irregularity in mobility behavior of the simulated robots modeled as if they are operating in unstructured environment.

In our approach, we consider simple unknown world with geometrically identifiable obstacles. Construction of the world perception is achieved at three levels. Initially, fragmented sensory data gathered by robots are filtered, analyzed, and streamlined into a more geometrical representation, i.e., lines, and curves. Next, robots are localized through using a localization technique [15]. Figure 3 illustrates gradual world perception construction of a robot navigating a simulated indoor environment. Secondly, localization of robots within the environment is performed to localize the robot via a target tracking technique [16]. The refined world perception information along with estimated global positions of the located robots are registered in a lookup table for further processing. From the world perception information, we extract world geometrical features. A connectivity check is performed to

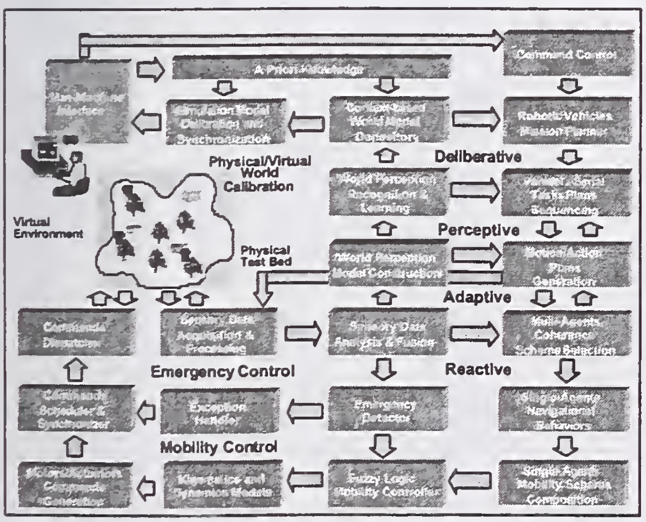


Figure 2. Modular Architecture of Supervisor Mobility Controller

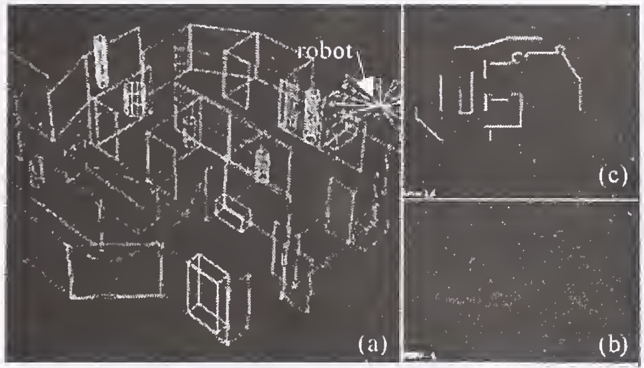


Figure 3. (a) A Simulated Robot World, (b) Reflection of Robot's Range Sensors Data as the Robot Navigates its Path, (c) Constructed Robot World Model Based on Sensory Data

identify neighboring line segments that closely represent a certain environment feature. The line segments belonging to a detected environment feature are marked accordingly along with a degree of certainty in another lookup table. To infer a robot's task plan and the world model, we use the latter lookup [16].

For deliberative task planning, cooperative robot team members must be planned based on a certain governing behavioral scheme. Some approaches in literature considered evolution of such governing behaviors as a way of demonstrating the cooperation among robots in an emergence way [10]. Some others attempted to motivate robots into cooperation [6]. While others have proposed consenting agents with mechanism for negotiation among cooperative agents [11]. For practical purposes, algorithmic techniques have shown good stability and performance [12,13]. We have developed different cooperative robots behavior-based algorithms for different applications [19-21].

Supervisor supports different multi-agent robot platform, (see Figure 4) and a number of cooperative navigational deployment strategies. Each robot by itself can be assigned variety of distinct behavior-based navigational schemes while engaging in a cooperative task. Each subsumption-based navigational behavior controls interaction of a robot with the environment and arrives at common decisions regarding turning and steering requirements of the robot for a given situation.

Supervisor handles task planning of cooperative robots using a high-level object-oriented language that will be discussed shortly. Supervisor recognizes two groups of task instructions. The first group of task instructions deals with mobility requirements of robotic vehicles and the other group deals with sensory requirements of cooperative robots (i.e., an instruction that activate the robot to perform an environment sensing operation).

Supervisor handles the Deliberative tasks with the highest priority while paying coarse attention to detailed of mobility of robots. In coordinated task planning of cooperative robots, coherence task planning of cooperative robots is very critical.

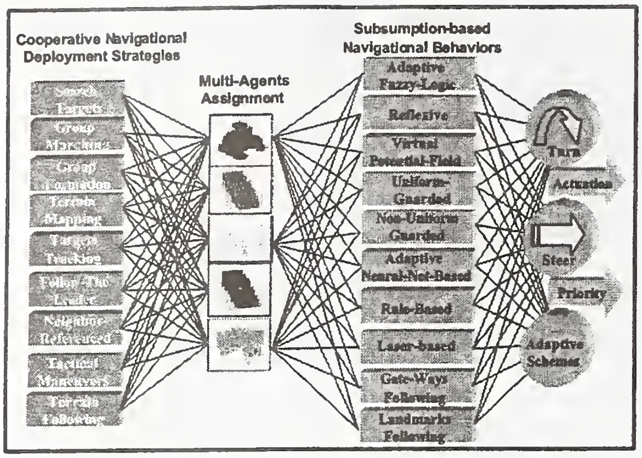


Figure 4. Cooperative Behaviors and Behavior-based Navigational Alternative of Multi-Agent Robots Supported by the Supervisor Mobility Controller.

Equally important, is the interaction between deliberative and reactive behavioral arbitration of cooperative robots. At the highest level of task planning, the mission plan can be designed using structured syntax and semaphore directives. Refer to Figure 5, for an example of cooperative robot task programming scheme. Supervisor has a parser that interprets abstract textual task commands as shown in the example. A double linked list buffers all task instruction with a time stamp.

```
Def Task1
Channel: 1,2,3                         // Assign Channel 1,2,3 For Comm
Vel: 5,5,5                  // Assign Velocity 5 in/s to Robots 1,2,& 3
Acc: 3,3,3                 // Assign Accelecration in/s2 to Robots 1,2,3
Loc: (10,30),(40,30),(60,-20)             // Assign 3 Vectors for the
                                                // the robots to follow.
NavBeh: 1,2,1             // Set Navigation Behaviors of Robot 1, 2,
                      //& 3 to Navigational Id # 1,2, 1 Respectively.
MC: 1                           //Turn on Continuous Processing Mode
Go: 1,1,1                       // Execution queued motion commands
                                              // for Robots 1, 2, and 3.
Wait t >4000                      // Wait until 4000 ms time is elapsed.
CapImg: 1,,1                       //Have Robots 1 and 3 capture images
                                                  // of their direction.
LMScan: ,1,                              //Have Robot 2 Scan Using Laser
                                          //Measurement Range Finder.
Go: 1,1,1           //Now, have all 3 robots to perform their sensing.
WaitAck: 1,1,1                  //Wait Until All Robots Acknowledge
                                   //Completion of Their Specific Tasks.
MC: 0                          //Turn off Continuous Processing Mode
EndDef                                            //Terminate Task Block.
```

Figure 5. An Example of Task Programming of Cooperative Robots.

For synchronization purposes, all motion commands and sensory data acquisition operations are buffered in a temporary transit queue until a trigger statement is executed. Execution of buffered commands is performed in the order that the commands have been received. In the example below, for instance, some preparatory commands set up communication channel, and preset velocity and acceleration of robot. The command *Loc* assigns three positional vectors to the robots to follow. The parameters of the *Loc* function define position and heading angle requirement of corresponding robot in inch and degree respectively. Next, proper navigational behaviors are assigned to individual cooperative robots. Instruction *Go* causes execution all buffered task plans to begin with the predefined task requirements, i.e., robots should presume a velocity 5 in/sec and an acceleration of 3"/sec^2. With continuous processing mode on, execution of commands proceeds right after executing Go statement without any delay. Next, the task command processing is delayed for 4 seconds before robots 1 and 3 are assigned to capture image along their heading direction, while the robot 2 is assigned to scan its heading for detection of obstacle. The last Go statement causes transmission of proper sensory data acquisition instruction to robots.

3. CONTROL SCHEMES FOR TASK PLAN TESTING OF COOPERATIVE ROBOTS

In practice, a mission plan may comprise of many sub-task plans – requiring cooperative to do many tasks either in synchronization or independent but in harmony. Each sub-task plan may consist of many symbolic notions of activities. Figure 6 shows one such sub-task plan where four robots are employed to rapidly create a consolidated world perception of their unknown environment. In this scenario, each robot's behavior is set to be reactive. Without explicitly defining individual navigational task of each robot, we applied a simple deliberative learning strategy. In this technique robots are rewarded more for exploring unvisited area of the world. The Supervisor performs two operations in this scenario - gathering range data from individual robots and fusing the range data to

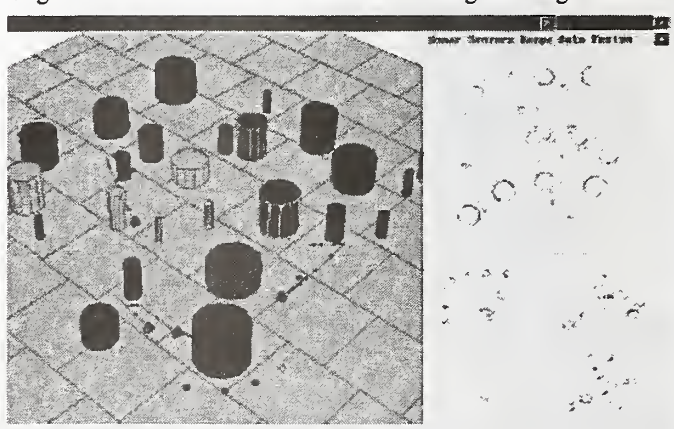


Figure 6. (a) A Simulated Scenario of Cooperative Robots World. (b) World Perception Model Based on Range Sensor Data Fusion of Cooperative Robots (c) World Perception of each of four individual Robots.

construct a world perception model as shown in the upper right hand corner of the figure 6. As the sonar range data become available, the world model is progressively constructed. The world model is partitioned into matrix of cells. Each cell will have a certain potential depending upon total number of sonar data point that it contains. The Supervisor creates a gradient field based on potential of each cell. The robots are then command to explore world environment with low gradient slope. To encourage the robots to explore the entire area, they are rewarded more for exploring the areas that they have not visited before. By adjusting the rewarding weights, the navigational behavioral of the robot are tuned. To prevent the robots from over exhaustion in their search, a time-based terminating condition is imposed that is if the new world discovery slows than behind a threshold over a fixed period of time, then the navigational search should stop. The algorithm was tested on a team of five cooperative robots. One robot from the group becomes as anchor and monitors operations of its other team members using its on-board surveillance camera. Localization of cooperative searching robots is performed using an image processing technique that localizes the cooperative robots in the image frame of the surveillance robot. The physical robotic test bed is shown in Figure 7. Coordinates of the robots in the image frame are next mapped to the world coordinate with the center of the camera at the origin [22]. A total of twenty tests were conducted to assess performance of the cooperative robotic team in detecting a total of 10 cans of cokes randomly located on the floor within an area of 20'x20'.

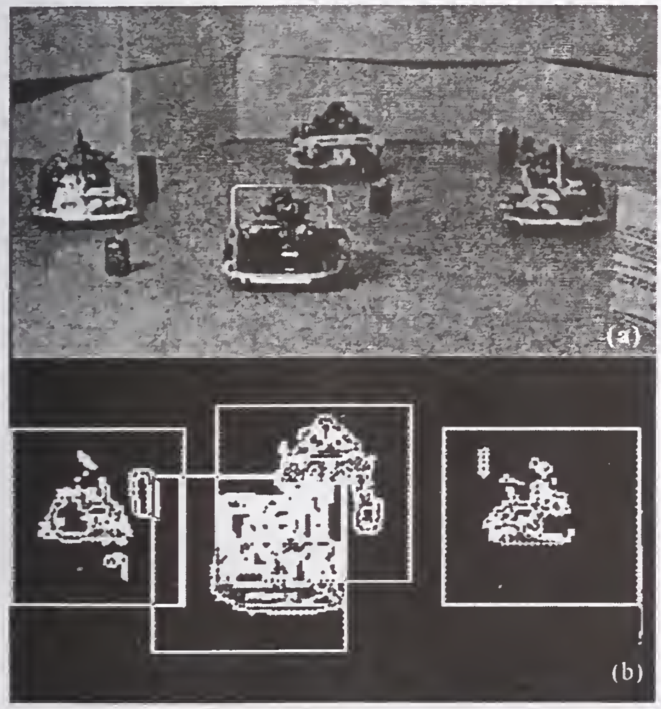


Figure 7. (a) Physical Cooperative Robot Test Bed. Localization of Cooperative Robot Using Visual Servoing Technique.

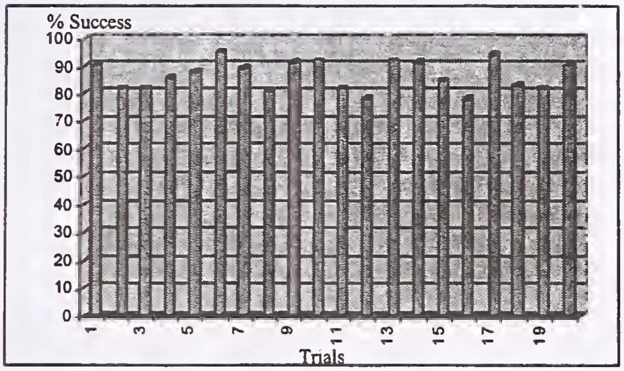


Figure 8. Performance Measure oCooperative Robots in Detecting Random Targets.

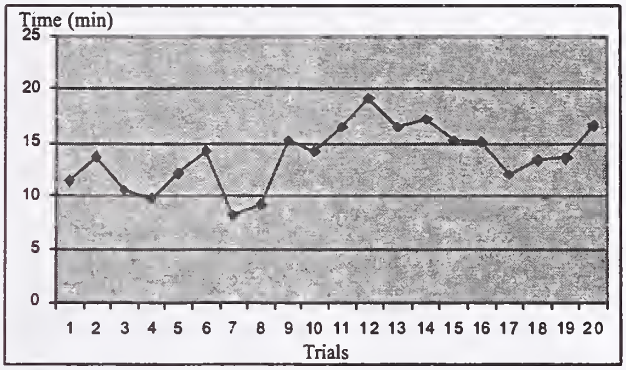


Figure 9. Time Efficiency of Cooperative Robots in Detecting Random Targets.

Figure 8 shows the results of experimentation. On average over 86.5 percent of time, all 10 targets were located with 99 percent confidence level. Figure 9 indicates the total time efficiency of the system in detecting all targets. While time efficiency of the system can serve as a basis for evaluation of intelligence of the robots, but some small adjustment in the control behaviors of the robots can have significant effect on overall efficiency of the system. Perhaps for reducing wondering time delay of the robots.

4. CONCLUSION

Cooperative robots have many practical applications. In this paper, we have discussed architecture of a Supervisory Mobility Controller with capability to facilitate deliberative/ reactive task deployment of cooperative robots. Some of the issues regarding robot's intelligence requirement at robot platform level and at cooperative robots level were discussed. To have fully functional cooperative robots, many research issues need to be addressed and taken into consideration. At present time, there is no single established standard or in testing procedures of the cooperative robots' intelligence. Measure of performance and efficiency of intelligent robotic system are very subjective and conditional. Minute adjustments in control parameters of a system can have significant influence over performance and efficiency of the system. Furthermore, incompatibility and heterogeneity among the robots makes it quite difficult to relate performance and efficiency of one system to another - even from one robot to another in the same platform and ranking.

5. ACKNOWLEDGEMENT

This work was partly funded by the US Army TACOM under DAAE07-01-C-L-065 and Penn State Applied Research Laboratory Under Prime Grant Number DAAD19-01-1-0504 from the US Army Research Office. The support from both funding agencies is acknowledged.

6. REFERENCES

[1] Mataric, M., "Behavior-based Control: Examples From Navigation, Learning, and Group Behavior," J. of Experimental and Theoretical AI, 9, 1997.

[2] Brooks, R., "Intelligence Without Reason," Computers and Thoughts - IJCAI 91, 1991.

[3] Arkin, R. C., Behavior-Based Robotics. The MIT Press, Cambridge, MA, 1998.

[4] Cao, Y. U., Fukunaga, A. S., and Kahng, A. B., "Cooperative Mobile Robotics: Antenedents and Directions, Autonomous Robots, 4(1): 7-27, 1997.

[5] Mataric, M.J., "Issues and Approaches in the Design of Coolective Autonomous Agents. Robotics and Autonomous Systems," 16 (2-4), pp. 321-331, 1995.

[6] Parker, L. E., "Cooperative Robotics for Multi-target Observation," Intelligent Automation and Soft Computing, 5(1): 5-19, 1999.

[7] Pirjanian, P., and Mataric, M. J., "Multi-robot Target Acquisition Using Multiple Objective Behavior Coordination," in Proc. Of the IEEE International Conf. on Robotics & Automation, San Francisco, 2000.

[8] Shirkhodaie, A., Goetz R., Patkar, A., Mansour M. Aderogba, S., and Avalareddy N., "Behavior-Based Tactical Navigational And Mobility Control of Tandem Ground Robotic Vehicles", in the Proc. of SPIE Unmanned Ground Vehicle Tech. II, Orlando, FL, 2000.

[9] Shirkhodaie, A., Rangavajhala, S., "Supervised Tactical Mobility Behavior Modeling And Control of Multi-Agent Robotic Vehicles," in the Proceedings of SPIE, Unmanned Ground Vehicle Technology II, April 24-25, 2001, Orlando, FL.

[10] Liu, J., and Wu, J., Multi-Agent Robotic Systems, The CRS Press International Series on Computational Intelligence, 2001.

[11] Rosenschein, J. S., "Consenting Agents: Negotiation Mechanisms for Multi-Agent Systems," in Proc. of Int. Joint Conf. On AI, pp. 792-799, 1993.

[12] Miyata, N. Ota, J., and Aiyama Y., Asama, H.., and Arai, T., "Cooperative Transport in Unknown Environment - Application of Real-time Task Assignment," Proc. of the 2000 IEEE Int. Conf. On Robotics and Automation, pp. 3176-3182, 2000.

[13] Mansour, M., and Shirkhodaie, A., "Navigational Behaviors Fusion for Multi-Agent Intelligent Tandem Mobile Robots," in the 2000 Japan-USA Symposium on Flexible Automation International Conference on Manufacturing Systems" Innovation for the 21st Century, Ann Arbor, MI, July 23-26, 2000.

[14] Lacaze, A., "Hierarchical Architecture of Coordinating Ground Vehicles in Unstructured Environments," in Proc. of 2001 PERMIS Workshop on Measuring Performance and Intelligence of Systems, September 4, 2001.

[15] Hajjawi, M., and Shirkhodaie, A., "Comparative Visual Team Working and Target Tracking of Mobile Robots", in the 34th IEEE Southeastern Symposium on System Theory, March 17, 19, 2002.

[16] Rangavajhala, S., and Shirkhodaie, A., and "Visual-Servoing and Navigation of Mobile Robots For Target Tracking Applications With Laser Sensory Data Fusion," in the Proceedings of the IEEE SSST 33rd Southeastern Symposium on System Theory, March 18-20, 2001.

[17] Nolfi, S., Evolutionary Robotics: Exploiting the Full Power of Self-Organization, Connection Science, pp. 167-183, 1998.

[18] Shirkhodaie A., "FMCell: User Manual - Vol I and II," Tennessee State University, Department of Mechanical and Manufacturing Engineering, April 2002.

[19] Adergoba, S., and Shirkhodaie, A., "Navigation and Visual-Terrain Tracking of Multiple Mobile Robots With Vision and Sonar Using Neural Network Techniques," in the IMCC'2000 International Manufacturing Conference in China," August 16-17, 2000, China.

[20] Avalareddy, N., and Shirkhodaie, A., "Communication-based Tactical Task Planning of Cooperative Tandem Mobile Robot Vehicles Under an Intelligent Supervisory Mobility Controller," in the 2000 Japan-USA Symposium on Flexible Automation International Conference on Manufacturing Systems", Ann Arbor, MI, July 23-26, 2000.

[21] Avalareddy, N., Shirkhodaie, A., "Cooperative Task Planning, Execution, and Monitoring of Mobile Robots Under a Supervisory Mobility Controller", in the IEEE Southeastcon 2000 Conference, Nashville, TN, April 7-9, 2000.

[22] Hajjawi, M.,"Visual Servo Tracking and Localization of Cooperative Robots," M.S. Thesis, Mechanical Engineering, Tennessee State University, August 2002.

Competitive relative performance evaluation of neural controllers for competitive game playing with teams of real mobile robots

A. L. Nelson and E. Grant
Center for Robotics and Intelligent Machines
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

T. C. Henderson
Department of Computer Science
School of Computing
3190 Merrill Engineering Building
University of Utah
Salt Lake City, Utah 84112

ABSTRACT

In this research, we describe the evolutionary training of artificial neural network controllers for competitive team game playing behaviors by teams of real mobile robots (The EvBots). During training (evolution), performance of controllers was evaluated based on the results of competitive tournaments of games played between robots (controllers) in an evolving population. Competitive tournament fitness evaluation does not require a human designer to define specific intermediate behaviors for a complex robot task. Intermediate behavior selection and evaluation becomes an implicit part of winning or losing games in a tournament. The acquisition of behavior in this evolutionary robotics system was demonstrated using a robotic version of the game 'Capture the Flag'. In this game, played by two teams of competing robots, each team tries to defend its own goal while trying to 'attack' another goal defended by the other team. Robot controllers were evolved in a simulated environment using evolutionary training algorithms and were then transferred to real robots in a physical environment for validation. Evolutionary robotics makes use of several distinct types or levels of performance evaluation. The work presented here focuses on the competitive relative tournament ranking metric used to drive the evolutionary process. After a population has been evolved, a second metric is needed to evaluate the quality of acquired game-playing skills. We use a post training evaluation method that compares the evolved controllers to hand coded knowledge-based controllers designed to perform the same task. In particular, a very poor controller, and high quality controller give us two points on a continuum that can be used to rank the evolved controller quality.

Keywords: *evolutionary robotics, performance metrics, mobile robot colonies, evolutionary neural computing, behavioral robotics*

1. Introduction

1.1 Evolutionary robotics

Evolutionary robotics (ER) is a relatively recent addition to the field of autonomous robot control research. ER focuses on the automatic design of model-free robot controllers using evolutionary computing methods. Over the course of last decade, proof-of-concept research in the field of ER has been conducted. Much of this work was done using computer-based simulations only [1][2][6]. Examples of ER research conducted with real robots include the evolution of walking behaviors in hexapod and octopod robots [7][8], and the evolution of simple behavioral controllers for small mobile robots[9][10]. The later include the development of phototaxis behaviors [11][12] and of simple object avoidance [10] and navigation [13]. For recent reviews of the field of ER see [14][15][16][13].

1.2 Intelligence performance metrics in ER

An ER application may make use of one or more types of fitness or intelligence metrics. These include 1) measurement of behavior quality for selection during training, 2) measurement of quality of transference from simulated training environments to the real world, and 3) post training evaluation of acquired behaviors. In this paper, we will focus mainly on development and formulation of the first type of metric. In addition, a post training metric will be applied to evaluate the quality of the best member of an evolved robot controller population.

In evolutionary robotics, a training performance fitness function is applied to provide selective pressure to an evolving population of robot controllers. The goal is to develop intelligent behavior with regard to a particular task. The nature and implementation of a machine learning application affects the way in which its intelligence can be measured. Behavioral robotics applications impose tight physical constraints on the expression and evaluation of learned behaviors. In particular, a subtle issue arises regarding the point of view of the intelligent robotic system and the point of view of the external observing human who is trying to evaluate that system's intelligence. These different points of view are known as the proximal (local) and distal (external) viewpoints respectively [3].

In most cases, fitness functions used in ER are formulated by designers from the distal point of view. Designers naturally develop fitness functions based on their own understanding of the desired behavior and system dynamics. In doing so, they implicitly incorporate information from their own complex distal world model into the fitness evaluation of the evolving agent(s). Since the evolving agent has no such model, the evolved behaviors tend to be very brittle. In the following section we will discuss some of the properties of such distal absolute fitness functions. As an alternative to absolute fitness functions we will present the formulation and experimental testing of an aggregate relative fitness evaluation method for ER.

1.3 Absolute vs. aggregate relative fitness functions in ER

Fitness in ER systems is often measured as an absolute scalar function to be maximized or minimized during training (see [4][5], for examples of this type of fitness function).

Absolute fitness functions used in evolving behavioral robotics applications are problematic for the following reasons: 1) Often, a forced learning plateau arises when the fitness metric is maximized or minimized, 2) Human biases are incorporation into the metric, 3) Each new robotic application requires a new and often difficult-to-formulate metric, and 4) For many complex behaviors, the knowledge required to specify an adequate absolute training fitness function is equivalent to that that would be required to design a rule/knowledge based controller by hand.

In this work we study a relative aggregate fitness selection metric for the evolution of behavioral robotics controllers. In particular, we focus on behaviors that can be formulated into competitive games played between two or more mobile robots. The metric produces a relative ranking in an evolving population of controllers, but does not give an absolute measure of fitness with respect to an external scale. Evaluation of evolving controllers based on their relative abilities to perform a task has the affect of aggregating evaluation of intermediate behaviors into one simple performance measure.

Tournament ranking evaluation partially eliminates the need to generate a fully domain-specific fitness function. As long as the problem can be formulated into a game that is either won or lost, other details about the game need not be included in the fitness function definition. Agents in an evolving population that receive higher relative rankings in a tournament of games will be propagated preferentially over agents receiving lower rankings. This reduces the amount of human bias that is incorporated into the performance metric. It also allows metrics to be specified in cases where humans lack adequate information to specify effective absolute fitness factors. This is important to the long-term scalability of ER methods to uncharacterized domains.

Implementing an aggregate competitive fitness function in the domain of evolving robot controllers is qualitatively different than similar implementations in pure computer science domains. In [17] tournament selection was applied to evolve neural networks to play computer Checkers with impressive results. In that work, the game board configurations were deterministically coded and fed directly into the neural networks. In ER, training environments must maintain an explicit I/O coupling analogous the robot's physical functional environment. This coupling must enforce a realistic proximal view onto the evolving agents. Modeled sensors must report only information that could be produced by the real sensors. Modeled motor actuators must in turn produce an alteration in the robot's frame of reference that appropriately alters the modeled sensor values (i.e. after the robot moves, it sees something new). This forms a controller-actuator-sensor loop with relational dynamics must be the analogous to those experienced by the real robots. The temporal fidelity of this controller-actuator-sensor loop must be maintained.

Another factor that complicates implementation of reinforcement learning of behavior in ER systems is that, many potential controller configurations may not lead to detectable expression of a desired behavior in a finite amount of time. In such cases, the search space must be restricted so that controllers will display detectable differences in performance even when that performance is measured at the aggregate level of win or lose. One way to do this is to formulate the competitive evaluation environment so that most controllers, even very poor ones, will eventually win at least a fraction of the games if their opponents are as poor or poorer than themselves.

2. The evolutionary robotics research testbed

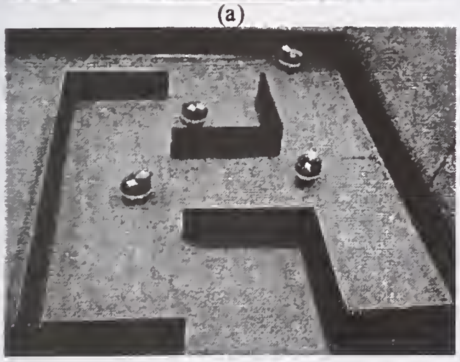
Before presenting a formulation of the relative aggregate fitness function used in this work, we will provide a brief summery of the ER research testbed used here. This will provide a context for the training fitness function formulation. The ER research testbed consists of a physical colony of autonomous mobile robots, and an evolutionary neural network training environment. The real robots use a vision-based range finding sensor emulation system to locate them selves in a physical maze environment. The evolutionary neural network training application uses simulation of the robots and their environment to evolve neural controllers to drive the robots. Controllers are evolved in simulation and then transferred to real robots for testing and verification.

2.1 The EvBot platform and environment

The physical verification and testing of evolved controllers developed in this research was conducted using a colony of small mobile robots named the EvBots (EVolutionary roBOTs)[18]. These robots are computationally powerful, fully autonomous and capable of performing all control, computing and data management on board. The robots move and steer using differential speed control of parallel drive wheels.

A physical maze environment was constructed for the mobile robot colony. Robots and objects in the environment were fitted with colored skirts to aid in vision based sensing of the environment. A fully assembled EvBot and the physical maze environment are shown in panels (a) and (b) of Figure 1 respectively.

(a)



(b)

Figure 1. Photographs of (a) a fully assembled EvBot and (b) the physical robot maze environment containing several robots.

2.2 Video range-finding emulation sensors

Each robot was fitted with a small video camera. Images captured from the video cameras are processed into object range date before being feed into the neural controllers.

The vision-based range-finding sensor systems on the robots used fixed geometric properties of the physical maze environment to calculate the ranges and angles of materials. Using color and position, the vision system could detect walls, robots, and goals. The goals are stationary cylinders and were used in the robotic 'Capture the Flag' game.

At each sensor update interval, and for each object type, range and angle values were calculated over the horizontal field of view of the robot's camera. A vector of range values was produced for each object type. Angular data was implicitly encoded in the order of the range values reported in each object range data vector. Object type information was not explicitly given to the robot neural controllers. Controllers were only given these resulting numerical data vectors. All associations relating distances, angles, and object types must be learned by the neural networks.

2.3 The evolutionary neural network architecture

In this research, a generalized evolvable neural network architecture capable of implementing a very broad class of network structures was used. The networks may contain arbitrary feed forward and feedback connections between any of the neurons in the network. Networks contain neurons with heterogeneous activation functions including sigmoidal, linear, step-threshold, and Gaussian radial basis functions. Neurons include a variable time-delay associated their inputs. This give networks the potential to evolve temporal processing abilities.

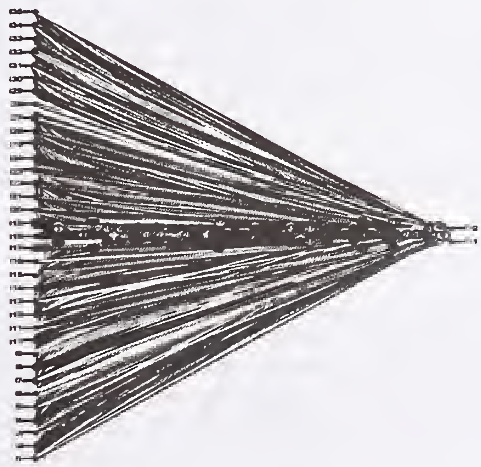


Figure 2. An example robot neural network controller from an evolved population of heterogeneous neural networks. The inputs to the network (left) are supplied by the robot's video range emulation sensors. The outputs of the network (right) are interpreted as wheel motor speed commands.

The connectivity and weighting relationships in a given network are completely specified by a single two-dimensional matrix **W** of real valued scalar weights. Additional Information specifying neuron types and time delays is given in a vector structure N with one formatted field per neuron. W and N form the basis of the genetic encoding for each network.

Figure 2 shows an example of a graphical representation of an evolved neural network.

2.5 Network mutation

The elements in the weight matrix and neuron information structures are acted upon directly by the genetic algorithm. Formally, the genome for a network **C** can be specified by the two dimensional matrix of real numbers

$$\mathbf{C} = [\mathbf{W} : \mathbf{N}'] \qquad (1)$$

where **N'** is a matrix of scalars extracted from the formatted structure **N**.

During evolution, networks are mutated in three ways. First, connection weight values can be perturbed. Second, connections can be added or removed. Finally, neuron units can be added or removed. Mutation of a network can be formalized by the compound relation

$$\mathbf{C}' = M_s(M_c(M_w(\mathbf{C}))) \qquad (2)$$

where **C** is the chromosome of the parent network and **C'** is the resulting mutated offspring network chromosome. M_w M_c and M_s are genetic operators that mutate the weights, the connections, and the neuron structure of the network respectively. Any or all of the different types of mutation can occur during propagation.

3. The fitness function and genetic algorithm

3.1 Fitness function formulation

In this section, we will define the fitness function used in this research. It is designed to be useful for team games that can be formulated to produce a win-lose outcome. These would include games like soccer and robot tag. Many useful real world behaviors such as mine sweeping and group searching behaviors in unknown terrain can be also formulated into scorable team robot games.

The training fitness function is comprised of two over-all parts: 1) select for controllers that win more games, 2) identify and select against pathological controller morphologies.

Only pathological cases that were known to lead to catastrophic stagnation of the evolutionary process were explicitly selected against. Two pathological controller behaviors were actively selected against. The first behavior was the production of constant continuous reverse wheel speeds in one or both wheels throughout the course of a game. The second pathological behavior was that of becoming stuck and remaining stuck for the duration of a game. These cases will be represented by Boolean functions *B1* and *B2* denoting the presence (1) or lack (0) of expression of each of the pathological behaviors, respectively.

A population **P** of evolving robot controllers consists of a fixed number *P* of neural networks. At the beginning of a tournament, a set of game starting positions for robot teams and goals is quasi-randomly generated and used for every game in that tournament (generation). In every training generation, a full tournament of games is played: Each controller in the population **P** plays against every other controller in **P**. After a tournament of games, each controller is given a score that depends on the number and quality of wins it achieved. For every pair of controllers in the population, two games are played. In the first game the first controller is used in the first team of robots and the second controller in the second team of robots. In the second game, the controllers are switched. This eliminates any advantage a controller may have incurred due to the random initial conditions used in the games of that tournament.

A generalized form of the fitness function for an individual controller after a tournament has been played can be written as

$$F(p) = w + d + n \qquad (3)$$

Where w, d and n are functions evaluating the contributions of games won, games played to a draw and expression of pathological behavior respectively during a tournament. $F(p)$ gives the relative fitness of the pth controller from the population **P**.

The relative fitness of the robot controllers playing in one game is dependent on the outcome of a reciprocal paired game in which the starting positions of the controllers are reversed. We will denote these paired games as g and g'. Using these paired games we break the game wins into three classes. In Class 1, a particular controller wins both games g and g'. Games of class 2 are those in which one controller wins one of the games but plays the other to a draw. In class 3, one controller wins one game but loses the other. Let *G1*, *G2*, and *G3* denote numbers of games won during a tournament of each of the three classes respectively. Then the number of points awarded to the pth controller for games won in a tournament is given by:

$$w = a * G1 + b * G2 + c * G3 \qquad (4)$$

Where $a, b,$ and c are scalar weighting factors. The values of a, b, and c are generally set so that $a>b>c>0$. This reflects the evaluation that a controller that can win from both of the starting positions of g and g' is better than one that can only win one of the two games.

Points are also given in the case that both of the games g and g' are played to a draw. If the robot agents of a particular controller are closer to their opponent's goal in both games, that controller is awarded points. The d sub-function of equation (3) becomes

$$d = d^*D1 \quad (5)$$

where $D1$ denotes the number game pairs played to a better draw and δ is a scalar weighting factor. δ is set to be much less than a, b, or c so that results related to numbers of wins dominate the tournament selection process.

Similarly, The n sub-function of 3 selecting against pathological behaviors can be expanded as

$$n = a^*B1 + \beta^*B2 \quad (6)$$

Where $B1$ and $B2$ are Boolean functions denoting the presence (1) or lack (0) of expression of each of the pathological behaviors in the current tournament (these were defined above as continual backward motion and becoming permanently stuck, respectively). α and β are scalar weighting factors and are generally set to be large negative values relative to a, b, and c so there is a heavy selective pressure against these behaviors even if they result in wins.

3.2. The evolutionary training algorithm

Populations of fixed size P were evolved using an evaluation, mutation, and replacement scheme. After each tournament of games, controller population members p were scored relative to each other using the performance metric $F(p)$ defined in equation (3). The population **P** was then reordered from fittest to least fit before propagation. The next generation population **P**_*next* was then constructed from the union of the following three sets derived from the current (parent) population:

$$\mathbf{P}\_next = \{p_1 \cdots p_m\} \cup \{p'_1 \cdots p'_m\} \cup \{p_{m+1} \cdots p_{P-2m}\} \quad (7)$$

Where $p_m \in \mathbf{P}$ is the mth individual of the ordered current (parent) population **P**, p'_m is a mutated version of p_m, and P is the fixed population size. Equation (7) produces a next generation composed of the following sets: 1) m of the fittest controllers are transferred un-changed to the next generation, 2) m of the fittest members of the controller population are mutated using equation (2) and added to the next generation, and 3) The remainder of the next generation population is made up of the remaining fittest remaining members of the current population. Although this algorithm is technically a form of greedy mutation-only (μ + λ)-ES with incomplete replacement [19], the game environment initialization for each tournament affects the outcome of the games to such a degree that the fittest member of the population could be eliminated. This adds a high degree of probabilistic selection to the algorithm.

4. Results and Discussion

4.1 The game

In this section, we present initial results and tests of one population of robot controllers evolved to play robot 'Capture the Flag'. In this game, there are two teams of robots and two stationary goal objects. All robots on team one and one of the goals are of one color (red). The other team members and their goal are of another color (green). In the game, robots of each team must try to approach the other team's goal object while protecting their own. The robot which first comes within range of its opponent's goal wins the game for its team.

Evolved controllers were able to play and win games both in simulation and when transferred to real robots in the physical world. The best evolved controllers acquired several distinct and testable abilities. These included avoidance of ones own goal, wall avoidance, goaltending, blocking and chasing robots from the other team, and homing in on an opponent's goal. Evolved controllers generally acquired two or three behaviors and exploited those rather than developing many behaviors for individual situations.

4.2. Experimental setup

We will focus on an evolved controller that displays two identifiable sub-behaviors: wall avoidance and selective avoidance of the robot's own goal. The controller was evolved in a population of size P=6 for 366 generations. The population replacement rate was set to 50% per generation. The parameters relating to the performance metric $F(p)$ of equation (3) used in this training evolution are given in Table 1 below.

Table 1. Fitness function parameter settings used to evolve the controller studied in these experiments.

Parameter	Game case (g, g')	Points awarded
a	win-win	20
b	win-draw	15
c	win-lose	10

δ	best draw	2
α	backward	-10
β	stuck	-2

4.3 Performance of evolved controllers in the real world

Here we will present experiments aimed at measuring the quality, and indirectly the intelligence, of the evolved robot controller. These post-evolution evaluation experiments were done in the real world using real robots.

Two knowledge-base controllers were developed. The first was designed to be a difficult opponent to beat and made use of both temporal and spatial information to avoid walls, extract itself from corners, avoid team mates, block opponents and home in on the opponent's goal. The second controller was designed to be a poor player and produced random wheel speed commands at each time step. In both cases, sensor input and motor output ranges were restricted to those allowed in the evolved neural network controller.

The evolved neural controller competed in a series of 20 real games against both the good rule-based and the random controllers. This was done to rank the quality of the evolved controller on a continuum including a good controller and a very poor controller. All games were recorded by collecting sequences of video images from a camera mounted directly above the maze.

Ten games were played between the evolved neural network and the good rule-base controller. Game initial positions may give one of the teams an advantage. For this reason, the set of games contained two games for each starting configuration used. In the first, each team was in a particular initial position, and in the second, the two team's starting positions were swapped. For these 10 games, five initial game positions were generated based on the random seed states 11 to 15 of the MATLAB random number generator. The games were conducted completely in the real robot maze environment using real robots. No games were allowed to proceed for longer than 200 controller update cycles (time steps). In addition, games were terminated if all robots became permanently stuck.

A similar set of ten real games was also played between the evolved neural network and the random (poor) controller. Again, the same set of five game initializations was used to conduct a set of 10 paired reciprocal games.

Tables 2 and 3 give the results of the games involving the good knowledge based and the random controllers respectively. The evolved neural controller, the good knowledge based controller and the random controller are denoted as "neural", "rule" and "random" respectively.

Table 2. Results of the set of ten games played between the evolved neural network controller and the hand coded rule-based (good) controller.

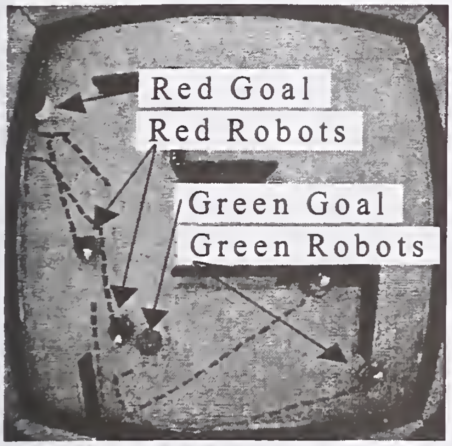
Game	Random Init. state	Team1	Team2	Winner
1	1	neural	rule	neural
2	1	rule	neural	rule
3	2	neural	rule	neural
4	2	rule	neural	rule
5	3	neural	rule	rule
6	3	rule	neural	rule
7	4	neural	rule	rule
8	4	rule	neural	neural
9	5	neural	rule	rule
10	5	rule	neural	rule

Table 3. Results of the set of ten games played between the evolved neural network controller and the hand coded random (poor) controller.

Game	Random Init. state	Team1	Team2	Winner
11	1	neural	random	neural
12	1	random	neural	neural
13	2	neural	random	neural
14	2	random	neural	(none)
15	3	neural	random	neural
16	3	random	neural	neural
17	4	neural	random	neural
18	4	random	neural	neural
19	5	neural	random	(none)
20	5	random	neural	neural

Summarizing these results, the neural network controllers won 3 out of 10 games against the good rule based controller, or 30% and the good rule base won 7 out of 10 or 70% of the games. All of the games between the neural network and the rule-based controller were played to completion. The neural network controllers won 8 out of 10 against the random controller, or 80%. The random controller was not able to win any games. In this case two of the games were not completed because all the robots became stuck or the game proceeded for more than 200 moves.

Figure 3 shows two example game results from the above tables. These are games 2 and 12 respectively. The robots are shown in their final end-game positions. The dotted lines indicate the courses of the robots during each game. In the first game (Figure 3 (a)), neural network controllers (green, lighter dotted lines) compete against good knowledge-based controllers (red, dark dotted lines). In the second game (Figure 3 (a)), neural network controllers (green) compete against poor random controllers (red). In the first game, the rule-based robots reach the green goal before the neural network based controllers can find the red goal. On the other hand, in the second game, the poorer random controllers are not able to make progress toward the green goal and the neural network based controllers eventually find the red goal and win the game.



(a)

(b)

Figure 3. Examples games played between trained neural network controllers (green robots, lighter dotted lines) and knowledge-based controllers (red robots, dark dotted lines). In (a) good rule-based robots beat neural network controllers while in (b) neural controllers eventually beat random controllers starting from similar initial conditions.

4.4 Discussion

These results imply that the functional quality of the evolved controller is somewhat less than that of the hand coded rule base. This is compared to the base line negligible abilities of the random controller. The evolved controller was able to beat the random controller in every game played to completion. It should be noted that identical or equally matched controllers would receive the same number of wins when competing against one another in a set of reciprocal games. For example, the rule-based controller would receive 5 out of 10 wins when played against a copy of itself, or 50%. Also, the rule based controller wins against the random controller 100% of the time (data not shown).

Evolved behavioral robotics control systems do not yet rival well designed sophisticated knowledge based controllers. Nonetheless, These results indicate that an evolve controller can beat a hand coded controller a fraction of the time.

The method of post-training controller evaluation does not influence the functionality of the relative tournament fitness function used during evolution of the controllers. This means it is possible to use post training fitness evaluation functions that are inadequate to select for the behavior, but can still measure behavior after it has been evolved. Also, we can evaluate the evolved controllers using human biases without imbedding such biased into the evolved controllers.

5. Conclusions and future research

In this paper a new evolutionary robotics testbed was described. A tournament training performance evaluation function was implemented. This fitness function was used to evolve controllers for teams of robots to play a benchmark competitive game, 'Capture the Flag'. The fitness function was not based on specific features of the game and could be used to evolve behaviors for other multi-robot tasks.

An evolved controller was experimentally tested using real robots in the real world. The evolved controller competed against a sophisticated hand designed knowledge based controller in a tournament and was able to win a fraction of the games.

This work will be extended by applying the competitive relative performance metric to other related mobile robot behaviors and by investigating the related training dynamics. We will investigate the possibility of improving training measures without adding more task-specific information. Alterations of the training metric could include the weighting of some tournaments more highly than others. It is also of interest to investigate the affects of game initialization on controller evolution. This work used random game initializations for each tournament. Another approach would be to select several game starting configurations and use only these. This method would run the risk of controllers learning environment specific behaviors that would not generalize well but could reduce the negative effects of poor game initializations that result in equal relative scores for all controllers and thus generate no selective pressure.

References

[1] F. Gomez, R. Miikkulainen, Incremental Evolution of Complex General Behavior, Adaptive Behavior, Vol. 5, pp. 317-342, 1997.

[2] J. Xiao, Z. Mickalewicz, L. Zhang, K. Trojanowski, Adaptive Evolutionary Planner/Navigator for Mobile Robots, IEEE Transactions on Evolutionary Computing, Vol. 1, no. 1 pp. 18-28, 2000.

[3] S. Nolfi, Evolutionary Robotics: Exploiting the Full Power of Self-Organization, Connection Science, Vol. 10, pp. 167-183, 1998.

[4] M. Quinn, Evolving cooperative homogeneous multi-robot teams, Proceedings of the IEEE / RSJ International Conference on Intelligent Robots and Systems (IROS 2000), Takamatsu Japan, Vol.3, pp. 1798 –1803, 2000.

[5] J. Kodjabachian and J-A. Meyer, Evolution and development of neural networks controlling locomotion, gradient-following, and obstacle avoidance in artificial insects, IEEE Transaction on Neural Networks 9(5) (September 1998)

[6] M. Potter, L. A. Meeden, A. Schultz, Heterogeneity in the coevolved behaviors of mobile robots: The emergence of specialists, Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 2001.

[7] D. Filliat, J. Kodjabachian, J.A. Meyer, Incremental evolution of neural controllers for navigation in a 6 legged robot, In Sugisaka and Tanaka, editors, Proc. of the Fourth International Symposium on Artificial Life and Robotics. Oita Univ. Press, 1999.

[8] N. Jakobi, Running Across the Reality Gap: Octopod Locomotion Evolved in a Minimal Simulation, Proceedings of the First European Workshop on Evolutionary Robotics: EvoRobot'98, 1998

[9] D. Floreano and S. Nolfi. Adaptive behavior in competing co-evolving species. Mantra technical report, LAMI, Swiss Federal Institute of Technology, Lausanne, 1997.

[10] D. Floreano and F. Mondada , Evolution of homing navigation in a real mobile robot. IEEE Transactions on Systems, Man, Cybernetics Part B: Cybernetics, Vol. 26, No. 3, pp. 396-407, 1996.

[11] N. Jakobi, P. Husbands, I. Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In F. Moran, A. Moreno, J. Merelo, and P. Chacon, editors, Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life, Springer-Verlag, Lecture Notes in Artificial Intelligence 929, pp. 704-720. 1995.

[12] Richard A. Watson, Sevan G. Ficici, Jordan B. Pollack, Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots, Robotics and Autonomous Systems, Vol. 39 No. 1, pp 1-18, 2002.

[13] I. Harvey, P. Husbands, D. Cliff, A. Thompson and N. Jakobi, Evolutionary robotics: the Sussex approach, Robotics and Autonomous Systems, Vol. 20, No 2-4, pp. 205-224, 1997.

[14] M. Mataria, D. Cliff, Challenges in evolving controllers for physical robots, Robotics and Autonomous Systems , Vol. 19, No. 1, pp. 67-83, 1996.

[15] S Nolfi, D. Floreano, "Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines", The MIT Press, Cambridge Massachusetts, 2000.

[16] D. Floreano, J. Urzelai, , Evolutionary Robotics: The Next Generation. In T. Gomi (ed.), Evolutionary Robotics III, Ontario (Canada): AAI Books, pp. 231-266, 2000.

[17] K. Chellapilla, D. B. Fogel, Evolving an Expert Checkers Playing Program Without Using Human Expertise. IEEE Transactions on Evolutionary Computation, Vol. 5, No. 4, pp. 422-428, 2001.

[18] John Galeotti, The EvBot A Small Autonomous Mobile Robot for the Study of Evolutionary Algorithms in Distributed Robotics, MS Thesis, North Carolina State University, 2002.

[19] I. Ashiru, C.A.Czarnecki, Evolving Communicating Controllers for Multiple Mobile Robot Systems, Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Vol. 4, pp 3498-3503, 1998.

Representing Ground Robotic Systems in Battlefield Simulations

M. Fields
U.S. Army Research Laboratory
Aberdeen Proving Ground, Maryland 21005-5066
mafld@arl.army.mil

ABSTRACT

As the Army continues to develop robotic systems for combat and combat support missions, it needs to develop representations of intelligent system performance for its battlefield simulation tools. These simulation tools differ considerably in their level of abstraction, flexibility and scale. Constructing the actual performance model requires the modeler to consider three factors: 1) the purpose of the particular simulation study; 2) the overall fidelity of the target simulation tool; and 3) the elements of the robotic system that are relevant to the simulation study. In this paper, we discuss a framework for modeling robotic system performance in the context of a battlefield simulation tool. We apply this framework to a model of the Demo III robotic system used in the OneSAF simulation tool.

1. INTRODUCTION

As the U.S. Army continues to develop concepts such as the Future Combat System (FCS) that include robotic assets, it needs to develop representations of intelligent systems for its battlefield simulation tools. This is a formidable task. Robotics systems currently under development range from man-portable systems to large tracked or wheeled vehicles. The level of control required for these intelligent systems ranges from full-time remote operation to intermittent supervisory control. The simulation tools themselves have different levels of fidelity, different time scales and different intended uses. These tools allow the technology developers, the analysts and the soldiers to experiment with robotic systems in readily available, re-configurable virtual environments. Technology developers can use simulations to investigate system design questions such as payload composition and placement, vulnerability and the appropriate sensor mix for autonomous mobility. The soldiers and military analysts can use simulations to develop Tactics, Techniques and Procedures (TTP) and requirements for robotic systems based on parametric studies involving key scenarios run over several terrain databases representative of the types of environments the robot is likely to encounter. Finally, well-designed simulations can be used identify critical near term technology problems and help prioritize research efforts.

The purpose of this paper is to present a framework for modeling intelligent systems that applies to a wide range of battlefield simulation tools and simulation purposes. Table 1 shows a breakdown of the types of models and simulations used to support weapon systems development and acquisition. The table gives a level of detail for the model and some examples of the types of evaluations and model outputs that can be expected at each level. In general, models that fall in categories near the top of the table represent systems more completely that simulations in categories near the bottom of the table. Traveling down the table, the size of the simulated world and the number of entities represented in a battlefield engagement increases.. The categories are somewhat artificial – there are models and simulations that fall somewhere between categories given in the table. Two of the battlefield simulation tools currently being used to examine robotic systems are the

Simulation Category	Level of Detail Modeled	Performance Data/Models Required	Type of Evaluation	Example Output
First Principal Physics	Physical processes	*Not applicable*	Design Feasibility	Electric Field Strength
Engineering	Components, Subsystems	Possibly Subcomponent level	Subsystem Performance	LADAR elevation map
One-on-One	Complete Weapon Systems	Component level	System Performance	Probability of successfully navigating a cross-country path
Few-on-Few	Small Military Units (Squads to Company)	Component level System level	System Effectiveness	Specific Exchange Ratio (SER) *Red losses caused by a specific blue system*
Force-on-Force	Large Scale Combat (Battalion or Higher)	System level	Combat Utility	Loss Exchange ratio (LER) *Ratio of red to blue losses*

Table 1 A hierarchy modeling and simulation tools used to support weapons systems development and

Combined Arms and Task Force Evaluation Model (CASTFOREM), and OneSAF. OneSAF has been used to support the Demo III robotics program. CASTFOREM will be used to provide weapon systems analysis for the Future Combat System program.

The U.S. Army Training and Doctrine command uses CASTFOREM to study force composition and system effectiveness at the Brigade and Battalion level. It is primarily a Force-on-Force event driven simulation. Processes such as detection and individual system damage are modeled stochastically using performance data provided for each weapon system. The actions of combat units are controlled by expert systems. Human participation is limited to preparing the data, rule sets for the expert systems and scenario design [1].

OneSAF is a real-time distributed interactive simulation tool developed by the U.S Army Simulation, Training and Instrumentation Command. It is used to training soldiers and to examine weapon systems concepts in brigade and below scenarios. It can be used to model engagements ranging in size from one-on-one encounter to battalion level exercises so it spans several of the categories given in Table 1. The actions of individual or aggregate units are controlled by behavior algorithms or human participants. Since it is a distributed simulation, it can be used in exercises involving different types of simulations, simulators and actual systems [2].

Constructing a particular robotic system model requires the modeler to consider three factors. First, it is important to keep in mind the purpose of a particular simulation study. Examining the contributions of a robotic scout to a battalion-level movement-to-contact scenario requires a much different model than examining the effect of a planning algorithm on autonomous driving. The overall fidelity of the model is also a major consideration. Higher fidelity simulation tools are compatible with physics–based models of robotic systems and subcomponents. Lower fidelity simulation tools use simple mathematical functions, often given as lookup tables, to represent subsystem performance. The quality of these lookup tables depends on the experimental data that can be collected for the robotic system being modeled. Finally, in constructing a model of the robotic system, a modeler needs to consider the elements of the robotic system that are relevant to the study. For instance, the overall performance of the driving sensor suite is certainly important for evaluating the contribution of robotic systems to a scout mission. The performance of an individual driving sensor may be less relevant.

In the next section, we present a general framework for robotic models that identifies the critical elements of a robotic system that need to be represented in any battlefield simulation. In the third section, we present some of the modeling and simulation tools that have been developed to support the Demo III robotics program. We also discuss how these tools can be used to guide the development of the robotic system performance models required by the Force-on-Force models like CASTFOREM.

2. A SYSTEM–LEVEL DESCRIPTION OF A ROBOTIC SYSTEM

It is useful to take a systems engineering approach and define a simulated robotic system as a collection of interlinked subsystems. It is important to note that this definition of the robotic system includes both the robotic vehicle(s) and the operator. Robots – even autonomous systems – cannot operate for long periods of time without human intervention. In terms of a battlefield scenario –the operator receives a mission and employees his robotic assets to complete the mission. The diagram given in Figure 1 shows a notional robotic system consisting of five major subsystems – Navigation; Communications; External Command and Control; Internal Command and Control; and the Payload System. Each of the major subsystems has elements relating to the mechanical and software components of the system.

In this notional robotic system, the External Command and Control System consists of the human operator, the man-

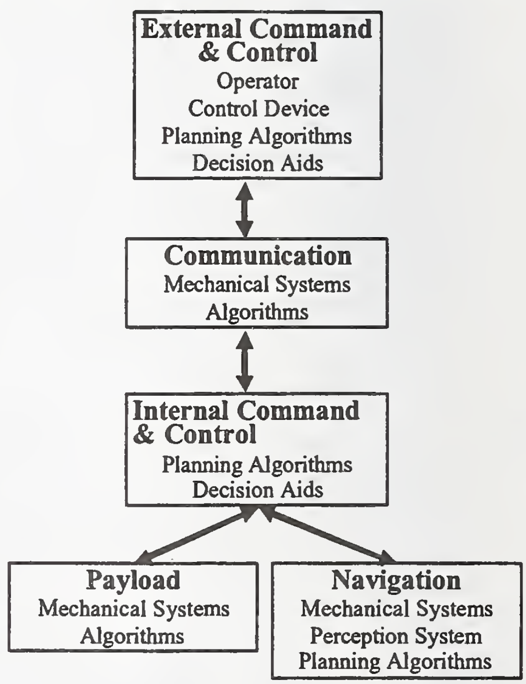


Figure 1. A systems engineering representation of an example robotic system

machine interface, and any command decision aids the operator may use. Depending on the application, this system could be located near the robotic vehicle or much further away. The Navigation System contains sensors and other hardware and software such as perception and planning algorithms. It resides on the robotic vehicle. The Communication System consists of the radios that link the human operator to the robot and the associated software. Components of this system are on the robotic vehicle and also co-located with the External Command and Control system. In one sense, the Navigation and Communications systems are support systems intended to allow the payload system on the robotic vehicle to contribute to the tactical mission. The Demo III robot carried a Reconnaissance Surveillance Target Acquisition (RSTA) package; other payloads such as weapons, storage containers, or smoke generators could also be represented. In general terms, the composition of the payload system consists of mechanical systems and supporting software algorithms.

The arrows in the pictures indicate the data flow in the system. The operator uses the communication system to give commands to the subsystems on the robotic vehicle. In this notional system, commands are passed through an intermediary command and control system that resides on the robot itself. All the subsystems interact with the simulated exercise, at least to some degree. They are subject to degradation or damage from elements in the simulated environment or from other entities in the exercise. Some systems such as the navigation system gather information from the environment to be used by systems on board the robot or at the external command and control station.

Diagrams similar to Figure 1 describe most weapon systems- we are using it to illustrate the types of models and supporting data needed to represent robotic systems in a battlefield simulation. There are two distinctions between robotic systems and most other weapons systems. First, mobility system performance must consider not only chassis response, but driver reliability, as well. Second, the robotic system is distributed. Many processes are semi-autonomous, requiring some level of operator participation. In a robotic system, time dependant performance measures such as "average target identification time" include communication time and two types of processing time– robotic processing time and human processing time.

For any study, we need to represent the performance of the major systems shown in Figure 1. Within engineering level simulations, the major systems may be represented by collections of high fidelity models of each of the processes contained within the system. Another engineering or quasi-engineering approach is to embed system components into a simulation tool. This allows researchers to "virtually" test hardware and software during the development process. As simulations become more abstracted, less detail can be included in the performance models. As researchers construct abstracted models of each of the systems, it is important to keep the following questions in mind. First, what is the purpose of the system? How reliably does the system accomplish its purpose? Finally, how quickly does it accomplish its purpose? The speed and reliability questions depend on collecting and analyzing experimental

A model of command and control must consider the types of decisions the operator makes, the speed of the decision process and the reliability of the decision process. Right now, data on the performance of the decision process is sparse. We can collect data from "virtual" exercises using embedded decision software or from field exercises.

To represent communications between the operator and the robot in a large-scale simulation, we need to measure the size and frequency of the messages. We also need to measure the speed and reliability of the system. We can gather some of this information from high fidelity communication models. Most of the information should come from integrated field experiments, where the robotic system has to accomplish mission similar to those used in combat.

Representing the navigation and payload systems also depend on gathering data to determine the speed and reliability of the process.

In the next section, we will describe some of our current models of the Demo III robotic system. Since the emphasis of the Demo III program is on developing autonomous mobility technologies, most of our modeling efforts have been directed at autonomous mobility as well. Many of the models we have developed describe processes within the navigation system. Recently, we have begun developing models for the other systems as well.

3. MODELING THE DEMO III ROBOTIC SYSTEM

Under the Demo III robotics program, the U.S. Army is developing a small survivable experimental unmanned ground platform (XUV) capable of autonomous operation on rugged terrain. Although the primary focus of the Demo III program is to develop and demonstrate autonomous mobility technologies, the research was focused on providing a robotic system for platoon level scout missions.

The Demo III XUV was designed in accordance to the NIST Real Time Control (4D/RCS) Reference Model Architecture which is a hierarchical structure designed to support the development of autonomous systems. Each level in the hierarchy is referred to as a node. A node consists of a behavior generation element, a value judgment element, a world model element, a sensory processing element and a knowledge database. The level of detail and dimensions of the "world" in the world model is a function of the node's position in the hierarchy – a node controlling several vehicles needs less resolution over a larger region to plan than a node that controls a single vehicle. Nodes receive goals, priorities, and plans from superiors and produce goals, priorities, and plans for subordinates.

The five levels of the 4D/RCS architecture are: Section Level, Vehicle Level, Subsystem Level, Primitive Level, and Servo Level. The section level receives a general plan generated at a higher level such at the platoon level. This plan contains a general command such as "Conduct a Tactical Movement" and a plan based on *a priori* information such as digital maps and situational awareness overlays. A section level plan is generally used to control multiple robots. At the Vehicle Level, the vehicle refines the Section Level command by developing a plan based on its world model which contains digital map data, situational awareness information and low-resolution information gathered by the on-board sensors. At this level, the vehicle refines the Section Level plan to avoid relatively large problem areas. At the Subsystem Level, the robot plans paths to avoid obstacles in its path. The Primitive Level controls the steering, acceleration, and braking of the robot. The lowest level in the architecture is the Servo Level - it controls the actuators for each of the subsystems.

Most of the modeling work for the Demo III system has focused on small scale battlefield experiments and engineering level studies. The primary purpose of these models has been to support the system design process. Systems are represented by collections of models for each of the major systems given in Figure I. As the technology matures, the community can begin to develop system level performance models. The challenge is to capture the system characteristics contained in the current collection of engineering and quasi-engineering level models into one model or mathematical function representing each of major systems. In the rest of this section, we discuss some of the existing OneSAF models representing processes and sub-processes within the major systems. We are beginning examine performance models

One of the modeling and simulation tools used to support the Demo III robotics program is OneSAF. It is a real-time battlefield simulation tool with a time step of ~0.015 seconds (66 Hz). It is an entity level simulation so all units are represented by collections of individual soldiers or vehicles. The baseline OneSAF represents hundreds of U.S. and foreign weapon systems and units. It has many pre-programmed behaviors to control the movement and interactions of those systems.

OneSAF is suitable for studying the interaction of robotic systems with other systems participating in small battlefield engagements. Because it is designed to interact with human participants, OneSAF is also appropriate for developing potential techniques, tactics and procedures for the use of robotic systems. However, because of it is a real-time simulation, it may not be appropriate for parametric studies requiring several replications. OneSAF's time step and battlefield environment are also too coarse for most engineering level studies. For example its environment is not detailed enough to be useful in evaluating the driving sensors, perception algorithms, or obstacle avoidance software involved in autonomous mobility.

3.1 Autonomous Mobility

The autonomous mobility model for the Demo III robotic system consists of the three main elements – a movement equation, a sensory processing suite, and a planning suite. The movement equation is a simple point model that determines the position, velocity, and acceleration of the vehicle at the end of each time step. The sensory processing suite builds a world model from inputs provided by the driving sensor suite. The planning suite uses the world model to determine a suitable path for the robotic vehicle. In the next couple of paragraphs, we describe the models that we used to represent each of these elements. In general, we can relate our modeling strategy to the 4D/RCS architecture. We can represent many of the processes at the Subsystem, Vehicle, and Section levels as algorithms that are executed in real time as a part of the overall simulation. However, We must depend on data and mathematical abstractions to represent processes on the Servo Level.

The time step for OneSAF is approximately 0.067 seconds. In this amount of time, the robot travels less than one meter (The maximum speed for the XUV is 40 kilometers per hours). We could excite the movement equation with sub-meter resolution terrain. Some high fidelity terrain databases for OneSAF are available, but they require large amounts of computer memory to use them efficiently. In our research, we use primarily 100m and 30m resolution terrain databases. We use a relatively simple equation of motion to model the motion of the XUV that uses the current position, velocity, acceleration and desired direction as inputs and gives the new position, velocity and acceleration as outputs. This equation is used in OneSAF to describe the motion of many of the ground vehicles.

Building a world model of the environment requires the driving sensor suite to gather information from the environment, process it, and present it to the planning suite in the form of a world model. The time step in OneSAF does not permit us to model the activities of the sensors themselves. Instead, we model the process of generating the world model from the simulated terrain database. In our simulation studies, we want the robotic vehicles to respond to relatively small obstacles such as woody vegetation and ditches that are not available on the *a priori* map. These are not features of a typical OneSAF terrain database. In our prior research (Fields, 1999), we developed techniques to increase add these features to existing OneSAF terrain databases. Figure N shows a section of a OneSAF terrain database with two types of mobility obstacles – positive obstacles shown dark gray and negative obstacles shown in light gray. Each of these obstacles is a polygonal feature with associated parameters used to specify a probability of detection function. Right now, detection of a particular obstacle depends on the type of the obstacle, its size (length, width, height), and the distance from the vehicle.

We produce two types of world models from the terrain database information. The first type of world model is a two-dimensional obstacle map with three types of pixels – clear, unknown, and blocked. Unknown pixels indicate areas within driving sensor range that are blocked from line of sight. Blocked pixels show the location of detected obstacles. Clear pixels indicate regions of the terrain that are visible to the driving sensor suite and free from detected obstacles. This is a useful representation of the obstacle detection process, but it is not the best representation of the world model used by the XUV. The XUV uses an elevation map to plan its near-term movements. Figure N shows a two-dimensional obstacle map and a three-dimensional elevation map for the same area. The heights in the elevation map are derived from two sources. The terrain skin provides the underlying ground plane elevation; detected obstacle polygons add or subtract elevation from this ground plane.

The planning process on board the vehicle consists of two planners – a near term planner operating at the subsystem level of the 4D/RCS architecture and a mid-range planner operating at the vehicle level of the architecture. In our work, we have developed two different models of the Demo III robotic planning process. In collaboration with the National Institutes of Standards and Technology (NIST) and Science and Engineering Services, Inc (SESI), we developed one model designed to examine the performance of the actual robotic planning software in tactical missions. This model requires software components both internal and external to the OneSAF simulation code. The actual vehicle level planner was linked to the OneSAF simulation code using the NIST neutral message language (NML) to pass plans from the planner to the simulated XUV. World models were passed from the simulated entity to the planner allowing it to use information gathered by the driving sensors on the simulated robot.

This same technique of linking actual software to the simulation system can be used to include the near-term planning system. In this case, the three-dimensional elevation map is passed to the near-term planner from the simulated world. Paths are passed back to the simulated entity.

The linked simulation is a good method to gather data on planning algorithm performance and to support the algorithm development process. We can experiment with the planners in different situations varying both the tactical situation and the obstacle distributions.

In the context of a larger exercise, possibly using another simulation tool, it may be impractical to link the actual code with the simulation. In this case, we want to use surrogate algorithms or mathematical models that perform similarly to the actual planning algorithms. We have used simpler algorithms to represent the near-term planning process. These algorithms use the two-dimensional obstacle map to plan the path of the vehicle.

3.2 The External Command and Control

The external command system for the Demo III robotic system consists of the operator, the operator control unit (OCU), and the associate planning software. There are two ways to represent the external command and control. The first method is to put a human operator in the simulation loop. The Mounted Maneuver Battlelab and SESI used this approach to support the Demo III program. The OCU was linked to OneSAF using NML to pass plans and other information between the operator and the simulated robotic entity. This approach of embedding hardware and software components into a simulation study allows researchers to collect data on operator activity and workload. Such information can be used to guide the design of effective control devices. Information from the embedded model also provides some system performance information that can be used to construct performance models for complex battlefield simulations.

We are beginning to construct an abstract model of the human operator. In its simplest terms, the human operator controlling one or more robotic assets is a server with a queue of heterogeous tasks to service. As with any queueing problem, it is the frequency and service times for each type of task that determines the workload on the operator. In our model, there are two types of service requests – mobility assistance requests and RSTA assistance requests.

3.3 The Communication System

We are beginning to address communication system models. Our approach is to model the amount of time required to transmit a message between the robot and the operator based as a function of message size. We are using this model in connection with our queuing theory model of the operator to introduce delays into requests for service and operator response time.

3.4 The RSTA Payload System

The RSTA system model uses existing models from the OneSAF simulation package. These models can represent many systems including camera systems, forward looking infrared devices, and radar systems.

4. CONCLUSIONS

In this paper, we presented a framework for developing models of intelligent systems that applies to a wide range of battlefield simulation tools and simulation purposes. The framework consists of five major systems – External Command and Control; Communications, Internal Command and Control, Navigation and Payload. Each of these systems needs to be represented in a battlefield simulation regardless of the level of simulation. In lower level simulations, we are able to use detailed models and/or components of the robotic systems to represent the robotic system. As the scale of the combat model increases, we need to develop abstract performance models of the systems within the robotic system. The validity of these performance models depends on the experimental data used to construct them.

5. REFERENECES

[1] Army Modeling and Simulation Resource Library .http://www.msrr.army.mil/index.cfm?RID=MNS_A_1000185, 1998.

[2] .

[3] J. Albus,"4D/RCS Reference Model Architecture for Unmanned Ground Vehicles", Proceedings of the SPIE Vol. 3693 AeroSense Session on Unmanned Ground Vehicle Technology, Orlando FL, April 7-8, 1999.

[4] M. Fields, "Modifying ModSAF terrain Databases to Support the Evaluation of Small Weapons Platforms in Tactical Scenarios", ARL-TR-1996, Army Research Laboratory, Aberdeen, Maryland, August 1999.

[6]W.Braudaway, C. Buettner, F Camberlain, A. Evans, M. Longtin, J. Smith, and T. Stanzione, "LibCTDB, Compact Terrain Database User Manual and Report",Naval Reseach Laboratory, Contract Number N00014-92-C-2150, 1996.

NICCI: A Multi-agent Cognitive Formation

Edward Dawidowicz
U.S. Army Communications-Electronic Command (CECOM),
Research Development and Engineering Center (RDEC)
Fort Monmouth, NJ
mailto:edward.dawidowicz@mail1.monmouth.army.mil

Keywords: *Network-centric, command and control, battlefield, behavior generation, entity-relational network, hierarchy of organization, knowledge, habitat, knowledge management, messages, and semantic network*

Abstract

The demand for effective and expediently made decisions is always in vogue. This is not surprising since making correct decisions is essential for successful operations in both the military and business environments. Decisions require data processed for quality, concept and context. The time is spent to weigh information for quality, to fuse information into concepts, and to package information for contextual relevance. The goal of information gathering and processing is focused on existing or arising problems. The network-centric paradigm allows for access to additional, previously unreachable, sources of information. While there is a benefit of getting more information, the time spent to weigh information for quality, to fuse information into concepts, and to package for contextual relevance is also increasing. Without a dramatic decrease in information-processing time, the network-centric paradigm[1] will not achieve its full potential.

The purpose of this paper is to propose a potential solution to the information overflow problem. The solution is proposed in a form of a system of cognitive agents, where cognitive agents are located at every information-processing nod. These cognitive agents we will call Intelligent Nodes. The Intelligent Nodes [4] *consist of machines and human-machine hybrids. Military processes are used in this paper to illustrate the application of a multi-agent cognitive framework as decision-maker assistants for making collaborative decisions better and faster.*

Introduction

The battlespace today and in the future is different from what we know based on the history of Twentieth century campaigns which manifested themselves as trench warfare of World War I and blitzkriegs of World War II. While the military is trained to fight twentieth century style campaigns, they find themselves today involved more and more in smaller conflicts where asymmetric warfare tactics are playing a bigger role. Stability and support operations (SASO) are the predominate type of operations conducted by the United States and coalition forces today.

With the emerging new technologies and tactics of this new type of warfare, 'bigger guns' do not always lead to a decisive victory and the need for a faster and more agile force has been recognized [3]. This new force, in addition to conventional weapons, will be armed with the latest information technology, where radios and computers are only part of the solution.

In this new warfare, the warfighter is no longer an isolated entity. The warfighter becomes a part of the operational environment where socioeconomic and physical laws play a greater role in the success or failure of a particular mission or even the war itself. The Network-centric warfare paradigm [2] is a new way of looking at this type of future battlefield. The Network-centric Infrastructure for Command, Control and Intelligence (NICCI) program is an attempt to find an out-of-the-box solution to an out-of-the-box problem.

Solutions to complex real world problems require looking at the problem space as a whole. If the solution is focused only on satisfying a few constraints, the solution will have weaknesses, which will be exploited by the opponent. The solutions emerge in a planning process as elements of a plan [1]. The incoming information influences and transforms earlier constraints, or produces new ones. This dynamic aspect of dealing with the real world complicates the planning process. Decisions are always focused on producing solutions which best satisfy the goals of achieving a certain effect based on current and anticipated

[1] Command and Control are responsible for making decisions in transforming goals into actions. The Intelligence supplies the information in regard to enemy positions and analysis of enemy courses of action (COA). In 2001, Dr. John Salasin has coined an appropriate name for the Joint DARPA program as Network-centric Infrastructure for Command, Control and Intelligence (NICCI).

conditions or states. The exchange of information relevant to the process therefore becomes very important.

NICCI introduces a novel concept called the habitat. The habitat in the conventional warfare schema is a set of combat, and combat support entities focused on achieving a given mission. Today habitats are more complex clusters of military and civilian entities; where information interchange is influenced by cultural, political, and socioeconomic realities. The military, paramilitary and civilian constituents of a habitat need a network-centric infrastructure to effectively communicate and collaborate in an attempt to focus in finding mutually benefiting solution. Due to the associated complexities, a multi-agent framework imbedded into the nodes of a communication network can assist members of a particular habitat with resolving issues such as information transformation, security, policy management and policy enforcement.

References

[1] A. Meystel, J Albus, "Intelligent Systems Architecture, Design and Control", John Wiley & Sons Inc 2002

[2] "Network-Centric Warfare: Its Origin and Future", Vice Admiral Arthur K. Cebrowski, U.S. Navy, and John J. Garstka Published January 1998, http://www.usni.org/Proceedings/Articles98/PROcebrowski.htm

[3] http://www.dtic.mil/jv2020/jvpub2.htm

[4], E. Dawidowicz, Albert Rodriguez, John Langston, "Intelligent Nodes in Knowledge Centric Warfare", Proceedings of 2002 Command and Control Research and Technology Symposium, Naval Postgraduate School, Monterey, CA, June, 2002

PART I
RESEARCH PAPERS

Plenary Lecture 1 - 1

Metrics and Performance Measures for Intelligent Unmanned Ground Vehicles

James S. Albus
Senior NIST Fellow
Intelligent Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
Technology Administration
United States Department of Commerce

Abstract

Metrics and measures for physical phenomena are precisely defined and widely accepted. However, metrics and measures for intelligent systems are as yet vaguely defined and controversial. Even the definition of intelligence is not widely agreed upon.

A number of metrics and measures have been developed for measuring human performance in scholastic aptitude, athletic ability, and task performance. Some of these suggest metrics and performance measures for intelligent machine systems. An example of a set of performance measures for unmanned military scout vehicles is presented.

Keywords: metrics, performance measures, unmanned ground vehicles, intelligent systems

1. INTRODUCTION

Webster defines a metric as a standard of measurement. Examples include the meter (a standard of length), the kilogram (a standard of mass), the volt (electromotive force), the ampere (electric current), the second (time), and degree Celsius, Fahrenheit, or Kelvin (temperature.) A metric is what is used to make a measurement. For physical metrics there is wide agreement on how the metric is defined and how it can be applied to precisely measure a physical entity or temporal event.

However, when we come to a metric for intelligence, there is much less agreement and much less precision. There is not even agreement on what intelligence is, much less on how to measure it, or even what is the metric for measuring it. Almost any meeting can grind to a halt over the attempt to define intelligence. I don't want that to happen here, so I am going to simply state my definition of intelligence, and move on.

Df: An intelligent system is a system with the ability to act appropriately in an uncertain environment

where

appropriate action is that which maximizes the likelihood of success in achieving or maintaining the highest level system goal. [Albus91]

The first thing to note about this definition is that intelligence has to do with a goal. Somewhere a goal is defined, and somehow the system accepts this goal as its own. The system then generates action that maximizes the likelihood that the goal with be achieved.

Note that the appropriate action may, or may not, be to head directly toward the goal. At the very lowest level, the proper action may depend not only on the current position relative to the goal, but on the current velocity and inertia of the system being controlled. At a higher level, a path planner may observe that the current vehicle position is in a cul-de-sac that blocks movement in the direction of the goal and plan a path away from the goal to escape the cul-de-sac.

Note also the reference to the "highest level" system goal. This implies a hierarchy of goals and subgoals with different planning horizons in time and space. At lower levels, goals are short-term, near-by, and high-resolution in time and space. At higher levels, goals are more distant and less precise in time and space.

Higher level goals may require that the system estimate the state of the world, gather information, build maps, plan routes, predict the future, imagine possible situations, weigh costs and benefits of alternative courses of action, and decide what behavior is most likely to achieve or maintain goals. For very high-level long-range goals such as rearing children, growing crops, or engaging in war, an intelligent system may need to make short term sacrifices, invent tools, develop weapons, and engage in deception.

2. LACK OF METRICS AND MEASURES

A major barrier to the development of intelligent systems is the lack of metrics and quantifiable measures of performance. There cannot be a science of intelligent systems without standard units of measure. To do science, you must be able to measure what you are doing and measure the results against some metric. This is something the field of AI, robotics, and intelligent systems has largely ignored. Most research results are in the form of demonstrations rather than experiments with data that that is quantitative and referenced against ground truth. There are few benchmarks or standardized tests wherein performance can be compared. That, of course is the subject of this workshop.

Perhaps the most common metric for human intelligence is the intelligence quotient (I.Q.) The average human I.Q. is arbitrarily defined as 100. But there is great controversy what I.Q. is and how it should be measured. There are, of course, many mental skills and abilities that can be measured. These include the ability to read and write, the ability to calculate with numbers, to reason with logic, to remember what was seen and heard, to perceive patterns, to understand relationships, and perform geometrical transformations. There are artistic skills and abilities. These include the ability to draw, paint, and sculpt, to sing and dance, to perform music, to compose poetry, to create or act out stories. There are manual skills and abilities to build or fix things, and athletic skills and abilities to compete in sports or fight in battles. In each of these areas, performance can be tested and scored.

Metrics for performance include speed (how fast?), precision (how accurate?), style (how graceful or well formed?), success/fail (criterion met?), effectiveness (desired result achieved?), efficiency (resources expended?), or cost/benefit (benefits worth the cost?)

Another type of performance metric is a benchmark. The performance of a system can be measured by comparing it against some benchmark performance. A benchmark may be an average over a population, or it may be a record of some kind, e.g., a world record, or a personal best.

One possible standard of measure is human performance. Human performance has been well defined and carefully calibrated in many areas. There are many existing measures of human performance, and human subjects are widely available. But how should this metric be applied? What should be measured? Perhaps the most fundamental measure is effectiveness in achieving goals. Certainly it is possible to measure timeliness. Cost, benefit, and risk are easily quantified for many tasks.

Typically a performance measure yields a score. The score may be some absolute quantity such as total points[1], or kilometers per hour, number of interventions per kilometer. Or the score may be a relative quantity such as order of finish in a race, or percentile in a distribution.

Sometimes the score takes into consideration the degree of difficulty of the performance. For example in competitive diving or figure skating, the performance score is multiplied by the degree of difficulty to decide the winner of a competition. In other cases, effort is made to assure that all competitors experience the same degree of difficulty – a so-called "level playing field." For example, in basketball and football games the teams switch goals at half-time. In races, all competitors are required to cover the same distance. In competitions where a level playing field cannot be achieved, there may be a preliminary competition to decide who gets the advantage. For example in automobile racing, qualifying time determines the starting line up. In tournaments, preliminary competition determines who meets the weakest competitor.

For autonomous driving, the degree of difficulty depends on the environment. On-road driving is more difficult on crowded streets and at intersections than on deserted roads and empty streets. The level of difficulty of off-road driving depends on the terrain and ground cover. It also depends on the density of obstacles such as ditches, trees, and rocks, and whether obstacles are hidden beneath tall grass or dense weeds.

In many cases, it is necessary to take into account the amount of training and preparation that have preceded the testing process. To accommodate these variations, different classes of competition may be established. Thus there are many issues with regard to how performance measurements should be made and how the results should be scored.

[1] where total points = points-per-goal x number of goals

Finally, there is the issue of what the score means. Typically the competitor with the most total points wins,[2] and rank order is determined by the number of points scored. However, not all competitions mean the same, and all wins or losses are not the same. Winning a pre-season game is not the same as winning the Super Bowl. In measuring the performance of intelligent systems, not all tests are equal. Passing a routine drivers test is not the same as qualifying for the "Indianapolis 500."

3. PERFORMANCE MEASURES

To address the issues of performance measures for intelligent systems, NIST has begun work in three areas:

1) a test course for search and rescue robots,
2) a measurement procedure for evaluating run-off-road detectors, and
3) a set of performance measures for autonomous driving.

The NIST test course for urban search and rescue addresses the problems of searching for human victims in buildings that have collapsed because of earth quakes, terrorist attacks, or other disasters. The USAR test course has been used in several AAAI and RoboCup competitions around the world. [Jacoff et al.00, 01, 02] This work is sponsored by the DARPA Mobile Autonomous Robot Software (MARS) program.

The NIST measurement procedure for run-off-road detectors addresses the problem of evaluating commercial products for effectiveness in determining when a vehicle is in danger of running off the road and warning the driver in time to prevent an accident. [Szabo et al.99] This work is sponsored by the Department of Transportation Highway Safety administration.

NIST work on metrics, performance measures, and standard reference data for autonomous driving addresses both off-road and on-road applications. This work is sponsored by the Army Research Lab Demo III Experimental Unmanned Vehicle (XUV) program. The Army is interested in measuring the state of readiness of autonomous driving technology for unmanned military vehicles. [Bornstein02] Specifically, the tests are designed to determine whether the Demo III XUVs have achieved technology readiness level six (TRL-6). TRL-6 requires that a prototype be demonstrated in a relevant environment.

For autonomous mobility, we assume that the relevant environment includes driving off-road through tall grass, weeds, and brush; through woods and fields, in desert and mountain terrain. The relevant environment also includes driving on-roads of all types including overgrown dirt trails, gravel roads, paved rural roads and highways, as well as and urban paved streets and alleys that may contain piles of rubble, burning tires, and abandoned vehicles. Relevant environmental conditions include day, night, rain, dry, dust, smoke, mud, and possibly snow and ice. It will not include the ability to

[2] except in golf where the competitor with the lowest score wins

autonomously cope with on-coming traffic, pedestrians, animals, moving vehicles, intersections, traffic signals, or road signs.

3.1 TRL-6 Test Procedures

The TRL-6 tests will proceed as follows:
For a chosen set of missions in a variety of environments (woods, fields, roads, trails, urban, desert, mountains) and a variety of conditions (day, night, dry, wet, snow, mud, dust, smoke):

- A manned scout vehicle will perform an assigned mission and its performance will be measured and scored for military effectiveness in terms of mission success, timeliness, resource expenditure, risk, and probable human casualties.
- A robot scout vehicle will perform the same mission and its performance will be measured and scored by the same criteria.
- The performance and score of the robot scout vehicle will be compared against that of the manned scout vehicle.
- A terrain characterization vehicle will exactly retrace the routes traversed by both manned and robot vehicles and will characterize the difficulty of the terrain covered in terms of slope, roughness, soil mechanics, and ground cover.
- Terrain characterization will help to define the operational envelope within which the robot vehicles can be used effectively.

3.2 Terrain Characterization

At least one way to measure difficulty is to measure the surface attributes of the terrain. The following is a set of terrain measurement techniques proposed for the coming TRL-6 experiments.

There will be a baseline and advanced set of terrain characterization measurements.

1. The baseline measurements will consist of one or more human observers riding in a HMMWV and subjectively scoring the difficulty of the terrain.
2. A more advanced scenario will include roughness measurements from an inertial navigation system, a TV camera, and an instrumented bumper on the XUV. These measurements will be compared with similar measurements made by similar sensors on the manned scout vehicle.
3. Still more advanced scenarios will include measurements made from the terrain characterization vehicle by high precision high-resolution LADAR .
4. The most advanced scenarios will include measurements of soil mechanics made from the terrain characterization vehicle.
5. If necessary, additional terrain characterization data will be obtained from overflights using airborne stereo cameras or LADAR scanners.

The terrain will be characterized by the following method:

1. A terrain characterization vehicle (HMMWV) will be driven over the exact paths traveled by the manned scout vehicle and the XUV during their respective missions.

2. The paths chosen by both vehicles will be scanned at regular intervals by high resolution LADAR cameras. These range images will then be registered with color images from color cameras. Data from an INS system will measure accelerations produced by bouncing over the terrain. Data from an instrumented bumper will measure the strength of the vegetation being driven through. Instrumentation to measure soil mechanics may also be carried by the terrain characterization vehicle.
3. The point clouds from the high resolution LADARs will be stitched together and the terrain will be characterized in terms of slope, roughness, obstacles, and ground cover. The conditions during the respective missions will be characterized by one or more of terrain characterization measurements described above.
4. Missions will be run under a variety of conditions including different times of day and night, different lighting, wet and dry, and different amounts of smoke and dust. The terrain attributes and conditions will be combined to provide a measure of difficulty for each path.
5. The two paths selected by the manned and robot vehicles will be scored and compared to determine which was the "better" path for the mission.

The difficulty of the terrain will also be analyzed to characterize the operational envelop within which the robot vehicle can be expected to reliably perform.

3. 3 Scout Vehicle Test Scenarios

The performance of the manned scout vehicle will be measured by the following procedure:

1. The Test Director will give a human commander a typical scout mission to be accomplished. The human commander will issue orders to the driver of a manned scout vehicle (a HMMWV) via a radio operator. The commander will be located in a HMMWV following the scout vehicle at a prescribed distance. The commander will have a map display overlaid with the manned scout location, the command vehicle location, and mission objectives. The commander may or may not have visual contact with the scout vehicle depending on the terrain and the separation distance between commander and manned scout vehicle.
2. A human driver in the manned scout vehicle will drive the scout vehicle in a tactical manner to accomplish the assigned mission objectives.
3. The manned scout vehicle will be scored by conventional methods used for evaluating human scout performance.
4. The number and type of conversations between the scout vehicle, the radio operator, and the commander will be measured.
5. The work load on the radio operator and commander will be measured.
6. The bandwidth and total amount of communications between manned vehicle, the radio operator, and the commander will be measured.
7. The behavior of the human driver, commander, and ratio operator will be video taped
8. A panoramic camera mounted on the manned scout vehicle will be used to record the scenes encountered by the human driver.

The performance of the robot scout vehicle (a XUV) will be measured by the following procedure:

1. The Test Director will give the human commander of the XUV the same mission as given to the commander of the manned scout vehicle. The human commander will issue orders to the XUV via a radio operator. The commander will be located in a HMMWV following the XUV at a prescribed distance. The commander will have a map display with the XUV location, the command vehicle location, and mission objectives. The commander may or may not have visual contact with the XUV depending on the terrain and the separation distance between commander and XUV.
2. The Demo III autonomous mobility system will drive the XUV to accomplish its mission objectives. The XUV may, or may not, choose the same path over the terrain as the manned scout vehicle. Either the commander or radio operator can provide intermediate way points to help robot get to goal point. Only the radio operator can teleoperate the robot.
3. The XUV performance will be scored by the same methods used for evaluating the manned scout vehicle.
4. The number of interventions by the radio operator will be measured and the type of interventions will be classified and analyzed.
5. The work load on the radio operator and commander will be measured
6. The bandwidth and volume of communications between XUV, the radio operator, and the commander will be measured.
7. A trace will be kept of critical state variables and world model representations during and prior to operator interventions.
8. The behavior of the XUV, the radio operator, and the commander will be video taped
9. A panoramic camera mounted on the XUV will be used to record the scenes encountered by the XUV.

4. SUMMARY AND CONCLUSIONS

Metrics and measures for physical phenomena are precisely defined and widely accepted. However, metrics and measures for intelligent systems are as yet vaguely defined and controversial. Even the definition of intelligence is not widely agreed upon.

There are a number of metrics and measures that have been developed for measuring human performance in scholastic aptitude, athletic ability, and task performance. It is suggested that these may provide guidelines for developing metrics and performance measures for intelligent machine systems. An example of a set of performance measures for unmanned military scout vehicles is presented.

6. REFERENCES

Albus, J. S. (1991) Outline for a Theory of Intelligence, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 3, pgs. 473-509, May/June

Bornstein, J. A. (2002) "U.S. Army ground robotics research program", *Proceedings of SPIE Aerosense Conference Vol. 4715,* Orlando, Florida, 1-5 April, 2002

Jacoff, A., Messina, E., Evans, J. (2000) A Standard Test Course for Urban Search and Rescue Robots, *Proceedings of the Performance Metrics For Intelligent Systems Workshop*, National Institute of Standards and Technology, Gaithersburg, MD, August 14-16

Jacoff, A., Messina, E., Evans, J. (2001) Experiences in Deploying Test Arenas for Autonomous Mobile Robots, *Proceedings of the Performance Metrics For Intelligent Systems (PerMIS) Workshop*, in association with IEEE CCA and ISIC, Mexico City, Mexico, Sept 4, 2001

Jacoff, A., Messina, E., Evans, J. (2002) Performance Evaluation of Autonomous Mobile Robots, *Industrial Robot 29:3*

Szabo, S., Murphy, K., Juberts, M. (1999) The AUTONAV/DOT Project: Baseline Measurement System for Evaluation of Roadway Departure Warning System, *NISTIR 6300*, National Institute of Standards and Technology, Gaithersburg, MD, March

PART I
RESEARCH PAPERS

1P1 - Performance of Mobility Systems

1. A Simulation Framework for Evaluating Mobile Robots
   S. Balakirsky, National Institute of Standards and Technology
   E. Messina, National Institute of Standards and Technology

2. Evaluating the Performance of a Vehicle Pose Measurement System
   H. Scott, National Institute of Standards and Technology
   S. Szabo, National Institute of Standards and Technology

3. Performance Evaluation of Road Detection and Tracking Algorithms
   D. Dufourd, DGA/Centre Technique d'Arcueil
   A. Digalarrondo, DGA/Centre Technique d'Arcueil

4. A Platform for Studying Locomotion Systems: Modular Reconfigurable Robots
   Y. Zhang, Palo Alto Research Center
   C. Eldershaw, Palo Alto Research Center
   M. Yim, Palo Alto Research Center
   K. Roufas, Palo Alto Research Center
   D. Duff, Palo Alto Research Center

A Simulation Framework for Evaluating Mobile Robots

Stephen Balakirsky and Elena Messina
National Institute of Standards and Technology
Intelligent Systems Division
Gaithersburg, MD 20899-8230
Email: stephen@cme.nist.gov, elena.messina@nist.gov *†

Abstract

As robotic technologies mature, we are moving from simple systems that roam our laboratories to heterogeneous groups of systems that operate in complex nonstructured environments. The novel and extremely complex nature of these autonomous systems generates a great deal of subsystem interdependencies that makes team, individual system, and subsystem validation and performance measurement difficult. Simple simulations or laboratory experimentation are no longer sufficient. To assist in evaluating these components and making design decisions, we are developing an integrated real-virtual environment. It is our hope that this will greatly facilitate the design, development, and understanding of how to configure and use multi-robot teams and will accelerate the robots' deployment.

Keywords:

simulation, architectures, 4D/RCS, mobile robots, algorithm validation

1 Introduction

There have been many recent successes in the field of mobile robotics. These range from single robot systems such as MINERVA that has been designed to give guided tours of museums [8], Predator and Global Hawk that have been designed for military air applications, and Demo III [7] [5] and Perceptor that have been designed for military ground applications to multi-robot systems such as the multiplicity of robot teams involved in the Robocup soccer league [3].

As these systems become more complex and attempt to perform more ambitious tasks, the knowledge and resources (both hardware and software) that are necessary to make contributions to the field dramatically increases. As a result, informal (or formal) code sharing now takes place between many universities and research institutions. For example, the code used in the Robocup competitions is published and freely available to anyone who wishes to download and use it, and the mobility and planning software used by the Perceptor program is heavily based upon the code from Demo III. While this code reuse allows researchers to gain quick entry into the various mobile robot arenas, it raises some interesting questions. If multiple sources of algorithms that provide a solution to a particular problem exist, which one is better? Will the given algorithm work in the new proposed environment? Are there any unintended consequences on the rest of the system (or systems) of integrating in this new component?

Code and component sharing also allows researchers to perform research in a specific area of robotics without becoming an expert in every aspect of robotics. For example, it should be possible on today's computer hardware to develop a planning system capable of creating plans to navigate through complex city traffic. However, in order to do this, an image processing system must exist that can detect and predict the location of traffic, road lanes, and traffic signs. It is feasible that a planning researcher can reuse a base technology such as the image processing subsystem rather then creating one from scratch. However, is there a solution to this problem when no such base technology or algorithm is available? Will the new

*No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipment, instruments, or materials are identified in this report in order to facilitate understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

†This work was sponsored in large part by grants from the Army Research Laboratories and the DARPA Mobile Autonomous Robot Software Program.

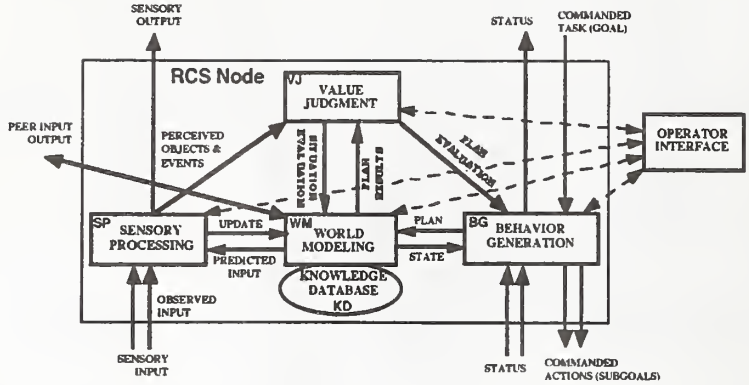


Figure 1: Model for RCS Control Node.

planning approach have to wait to be proven because of dependencies on as yet undeveloped algorithms or hardware?

This paper will suggest a means to address all of the above questions by describing the development of an integrated real-virtual simulation environment. The objective of this environment is to provide a standard architecture and set of interfaces through which real and virtual systems may be seamlessly coupled together. It will be shown that through this coupling, components ranging from individual algorithms to groups of vehicles may be developed, debugged, and evaluated.

2 An Architecture for Intelligent Autonomous Vehicles

One of the key decisions to be made in building any kind of complex system is how to organize the hardware and software. The Demo III Program and some of the teams competing for the Future Combat Systems contract have selected the 4D/RCS reference architecture for their autonomous vehicles [1]. Rather than starting from scratch, the simulation framework will be built upon the existing 4D/RCS architecture and will take advantage of existing interfaces and components.

The 4D/RCS architecture consists of a hierarchy of computational nodes, each of which contains the same elements. Based on control theory principles, 4D/RCS partitions the problem of control into four basic elements that together comprise a computational node: behavior generation (BG), sensory processing (SP), world modeling (WM), and value judgment (VJ). Figure 1 shows the 4D/RCS control node and the connections between its constituent components. Figure 2 shows a sample 4D/RCS hierarchy for military scout vehicles.

3 Requirements for a Simulation, Modeling, and Development Framework

An architecture is a first step towards guiding and facilitating the construction and evaluation of complex single or multi-vehicle autonomous systems. Tools that help automate the software development and component integration are another important element. NIST has been working with industry, other government agencies, and academia to investigate tools to facilitate construction of the types of large and complex systems that will be represented in this simulation framework. We are developing a large-scale simulation environment that will enable us, along with others, to design the control hierarchy, populate the control nodes, run the system in simulation, debug it, and generate code for the target host. The development and simulation environments are closely tied to the eventual deployment platforms and are intended to be able to operate with a combination of real and simulated entities. The ability to enable human-in-the-loop testing and execution is also crucial, given the novel aspects of human-robot interactions.

A high-level list of the requirements for such a development and simulation environment has been developed to help guide its creation. The requirements are as follows:

- Full support of 4D/RCS architecture
- Graphical user interface for developing, integrating, testing, debugging the system under development

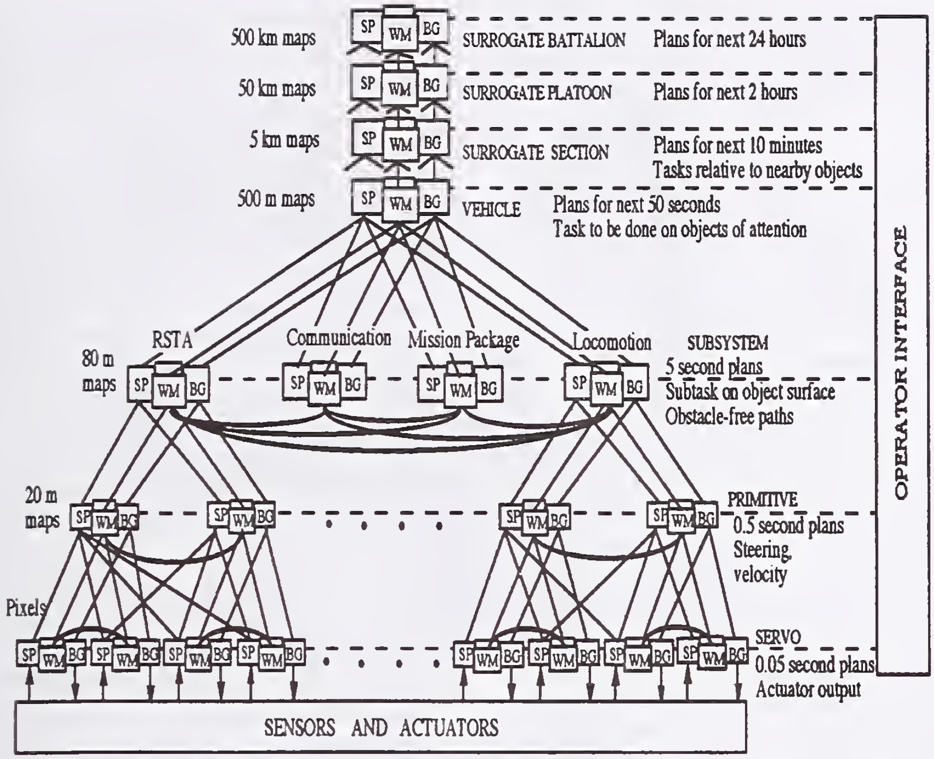


Figure 2: 4D/RCS Reference Model Architecture for an individual vehicle.

- Reuse support:
  - Architecture elements
  - Component templates
  - Algorithms
  - Code
  - Subsystems
- Intuitive visualizations of the control system to support design and development by providing an understanding of what the system is doing and why, and what it plans to do next. Examples of visualizations include:
  - Display of control hierarchy as it executes, including commands flowing down and status flowing up
  - Ability to "zoom in" on a particular node and view states as the system executes
  - Ability to view world models within the system
- Execution controls, including
  - Single step through execution
  - Breakpoints and watch windows
  - Logging
- Simulation infrastructure supporting realistic execution scenarios, visualization, and debugging/evaluation experiments. This includes
  - Population of the environment external to the system with relevant features (such as roads, other pieces of equipment, humans, etc.)
  - Controlled dynamic environment (with prescribed repeatable events)
- Modification capabilities so that the designer and user can perform "what if" experiments. The tools should allow interactive and intuitive modification of situations in the environment or within the system. The modification capabilities should work seamlessly with the visualization, simulation, and execution features. Examples of types of modifications that should be allowed include:
  - Changing world model data
  - Importing datasets that represent what the system's sensors would receive
  - Changing environmental conditions
- Support for real-time computing. All levels of the 4D/RCS control hierarchy must execute within

certain time constraints (e.g., the servo level may have to respond at 60 Hz; whereas higher levels may have several seconds or even minutes per cycle).

4 Proposed System of Systems

We are seeking to create an *integrated* environment that provides capabilities typically associated with software development tools and those associated with simulation environments. All of the pieces necessary for the construction of this environment may exist to some degree as separate commercially available packages. However, whereas several commercial tools exist to help design and construct software, these tool-sets are typically disconnected from the overall system architecture and execution environments (real or simulated). Likewise, many sophisticated simulation systems exist. However, these systems tend to work at either a very broad scope at low resolution, or a very limited scope at high resolution. What we are building is a coherent environment for designing, developing, and validating software components ranging in size from a single module to a team of systems.

As shown in Figure 3 one will be able to take software modules from a repository and interface them directly into either a real or virtual system. This ability will be supported through the decomposition of systems and algorithms into the 4D/RCS architecture and the use of standard interfaces between modules.

This decomposition will be supported through the software design and development support tools that provide the ability to work in a graphical environment to sketch the control hierarchy, bring up partially instantiated 4D/RCS control nodes, easily create connections between nodes (or components within them), and automatically generate executable code. The software support tools will also encompass capabilities typically found under run-time debug tools, including single stepping and setting break points. Furthermore, sophisticated displays of variables and execution states are being created. For instance, a "strip chart" view one or more variables can be displayed on screen and techniques for helping developers visualize complex world models are being designed. This includes display of the graphs that several path planning algorithms utilize to search for the best (least cost) path. The graphs typically have thousands of nodes, which are connected in a neighborhood to other nodes by edges that have a cost associated with them. The costs vary, depending on operation mode, or as environmental conditions change and are computed based on the various layers in the world model (such as roads, obstacles, and elevation). Therefore, the visualization of relevant and salient aspects of the world model in order to validate the model itself and the planning is a challenging undertaking.

The software development tools will segue smoothly into the simulation environment. Under this concept, a virtual world is being created that brings together existing multi-platform and single platform simulation systems into a system of systems. Through the use of well-defined interfaces that are supported on a wide variety of computer platforms, the simulator's internal command and data flows will be able to be interrupted and modified. This will allow researchers to "plug-in" individual technology components that meet the interface requirements and override the default methods that the simulators normally employ. As shown in Figure 3, interfaces will be provided that range through the entire spectrum of the 4D/RCS hierarchy; from a low-fidelity multi-platform configuration to a high-fidelity single platform configuration, to the ability to add real platforms into the virtual world.

Global variable resolution database resources will also be available. These include a terrain database that contains elevation data, a feature database that contains annotated vector data for roads, signs, buildings, rivers, etc., and an annotated entity database that contains information on all of the platforms participating in the simulation. The annotations include items such as lane markings and names for roads, text contained on a sign, and health status of other entities. Filters will be available to tune the database outputs to the specific needs of each algorithm. For example, specific sensor processing capabilities may be simulated by querying these databases with a specified sensor range and resolution.

In addition to serving *a priori* data, these databases will be able to be modified in real-time. Any modifications made to the databases will be viewable by all participants (both real and virtual) in the exercise. These modifications may be related to sensed information from a real vehicle that is participating in the simulation or may be injected by the user to alter the environment that the simulation framework in operating in.

The final component of the system is a set of data capture, analysis, and evaluation tools. The data capture tool will allow for any or all of the messages being transmitted over the standard message channels to be time-tagged and logged into a trace file. In addition, raw simulation results may be logged, for example the location and activities of each participating entity.

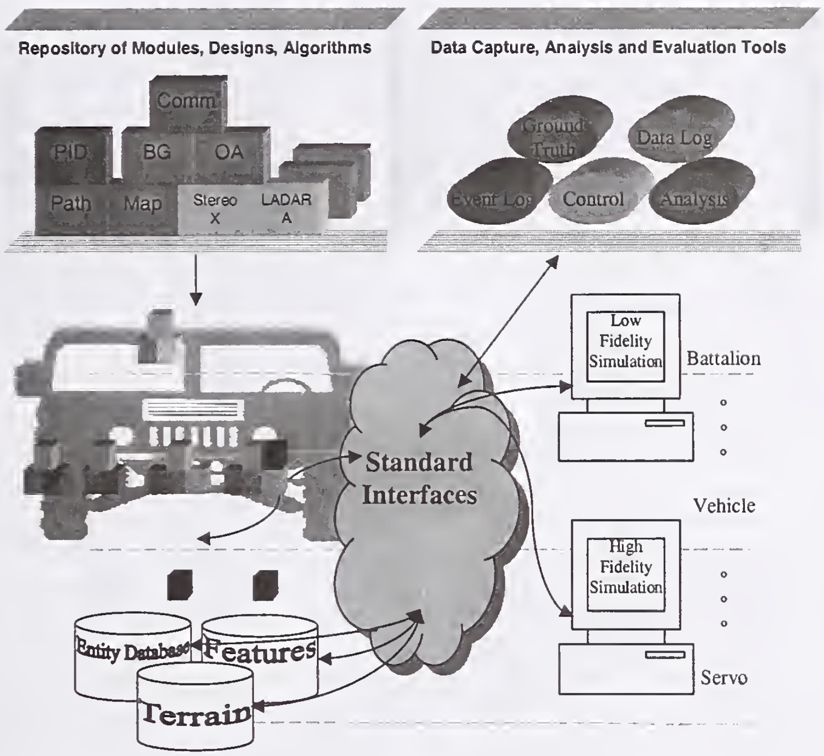


Figure 3: Hierarchy of simulators.

Data analysis tools are also being developed that will allow for a variety of data presentation and analysis options. This will include both the tracking of timings as well as the values of specific variables or combinations of variables. For example, the distance between two entities may be constantly tracked. These data analysis tools will also be tied into ground truth from the simulation systems. This will allow algorithm and system evaluation against known data and the determination of such items as the time from a sensor event to a system reaction or the accuracy of a real systems road following algorithms. Through the use of these standard interfaces and tools, researchers will be provided with a low cost technique for evaluating performance changes due to their system's or algorithm's integration into the overall framework.

5 Current Implementation

While the entire software development and simulation system has not yet been implemented, progress has been made on developing prototypes and designs for the overall system. No single simulation tool has been found that meets all of the criteria discussed in the requirements section. Therefore, a hierarchy of simulators has been explored.

At the top of the hierarchy, a low-fidelity, long temporal and spatial duration, multi-platform simulator is required. As designed, this class of simulator is capable of simulating the interaction, coordination, and movement of large groups of platforms. While these simulators do simulate down to the level of the individual platforms moving across the terrain, the terrain and mobility models are typically low resolution. Therefore, this class of simulator is best utilized in developing algorithms for group behaviors where precise platform mobility and sensing modeling is beyond the scope of the experiment. A second class of simulator has been identified for situations where precise modeling is required. These simulators will need to share interfaces with the low-fidelity simulator, and in fact may take commands from the low-fidelity simulators in order to precisely model one or more of the platforms involved in a particular exercise. The high-fidelity simulators will also be able to read the shared databases and construct simulated sensor output (or pass real sensor output) that may be used by external sensor processing algorithms. Complex, dynamically correct platform motion models and high resolution terrain information will also available at this level.

Interfaces will be inserted into each simulator that will enable the export and import of both world model and behavior generation information at each level of the 4D/RCS hierarchy. This will enable researchers to implement a particular group behavior or algorithm at a particular level of 4D/RCS. For example, a cooperative search algorithm could be implemented at the "section" level of 4D/RCS. The algorithm would receive its command input from the platoon level of the low-fidelity simulator and construct a plan based on information read from the terrain, entity, and feature databases. The planned course of action (individual plans for several platforms) would then be passed back into the simulator for execution. In this particular case, the plans could be passed either back to the low-fidelity simulator or to the high-fidelity simulator. In addition, one or more of the platform's plans could be passed to real systems. As designed, the source and destination of these plans and the data utilized to construct them is transparent to the planning system and totally controlled by the user. This will facilitate an environment where a researcher can simulate as many or as few subsystems and levels as desired.

Both a low-fidelity and high-fidelity commercially available simulation package have been selected and implemented into the prototype framework.

5.1 Low-fidelity Simulator

For the prototype system, we have chosen the US Army STRICOM's OneSAF Testbed Baseline (OTB)[1] for both the low-fidelity simulation and shared database server. We have worked closely with the Army Research Laboratory and Science and Engineering Services Inc. to install the standard interfaces. All of the interfaces communicate over NIST's NML communication channels [6] which provide a multi-platform solution to inter-process communication.

Distributed, shared databases are implemented as part of the standard implementation of OTB. We have added interfaces into the simulation system that allow for simple outside access of this information. These interfaces include hooks into the terrain elevation database, feature database, and entity database. Additional channels that tie the basic information contained in these databases to a full relational database for the storage of attribute information is currently being investigated.

For the terrain elevation database, both all-knowing (what is the elevation in this area, to this resolution) and modeled (what is the terrain map as modeled by vehicle x with its sensors) are available. Feature vector data is available on an all-knowing basis that may be filtered by distance from the vehicle so as to simulate what sensors perceive. In addition to the standard features that are modeled by the simulation system, simulated traffic signals and signs are being implemented. For entity data, filtered information (all friendly, enemy, detected, etc.) reports are available as well as event detections. Events currently supported include line crossings and anticipated line crossings with more to be added shortly.

In addition to the database access interfaces, we are able to interrupt the standard OTB command flow to inject our own plans. This has been demonstrated by having OTB section level plans sent out over an NML channel to a stand-alone vehicle level planner. The results of the vehicle level planner can then be executed on real robotic hardware, sent to a high-fidelity simulator, or sent back into the OTB simulator for execution.

Input from real robot platforms into the simulation environment is also supported. This interface allows a real robotic platform to influence OTB databases by continuously updating their own location as well as adding detected features and entities.

Work is continuing on developing further interfaces. These will provide further breaks in the OTB command flows that will allow for planning systems that compute group plans to be implemented and evaluated.

Another feature that is standard with the OTB distribution is a data logger. This logging facility logs all entity movements and events that occur in the simulation. Logging facilities to log message channel traffic are currently under development.

5.2 High-fidelity Simulator

For the high-fidelity simulation, SimRobot from the University of Bremen[2] has been selected for the prototype system. NML channels for low-level command input, and position output have been implemented. Currently, this simulator is only capable of simulations on a flat earth. Therefore, work is being performed to improve the simulators ability to operate in complex 3-dimensional terrain. Once this work is completed, the low-fidelity simulator will operate from the same terrain database as the low-fidelity simulator.

[1] http://www.onesaf.org/publicotb1.html

[2] http://www.informatik.uni-bremen.de/~roefer/simrobot/index_e.htm

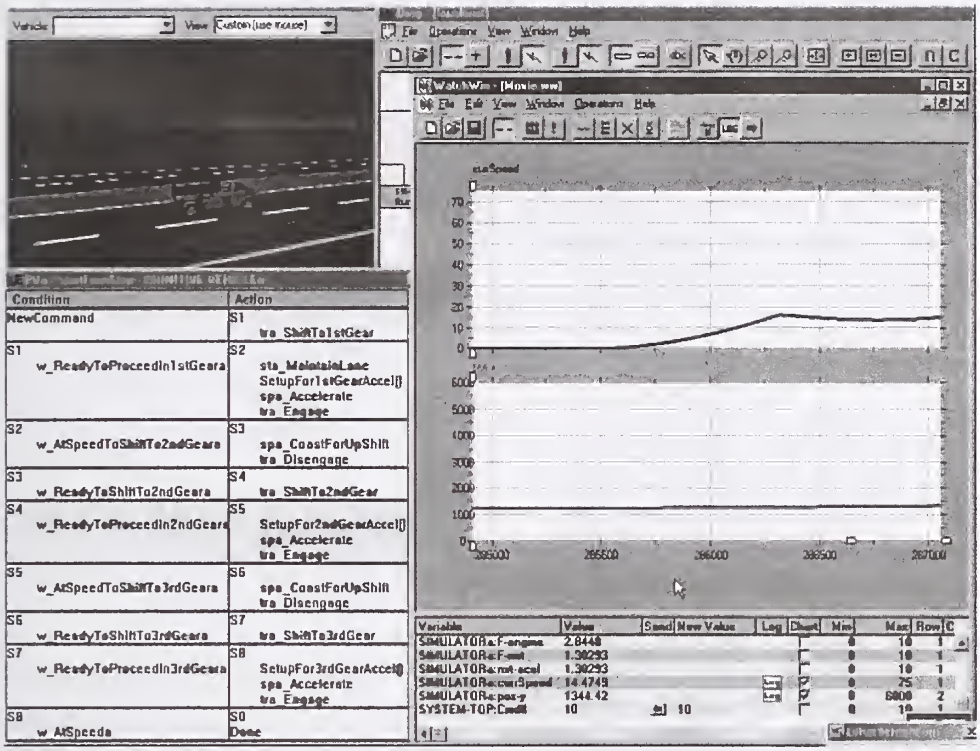


Figure 4: Example of features in controller software development tool.

5.3 Software Development Support

We have been experimenting with various representation techniques and development tools. These range from commercial packages, such as Real-Time Innovation Incorporated's ControlShell to novel formal languages, such as Stanford's Rapide [4]. Recent work has focused on the use of the Unified Modeling Language to support 4D/RCS control system development [2].

A commercial development and execution tool for building simpler versions of RCS-style controllers has been developed by a small company (Advanced Technology and Research), but it is targeted at manufacturing systems that have minimal sensing requirements. This tool is being modified to support the types of visualizations, modifications, and execution controls desired for on-road and off-road vehicles. Figure 4 is a screen shot taken of the tool while running and illustrates some of its features. The top left window shows an animation of the vehicle being controlled as it moves through its environment, which can include other vehicles. The top right window shows 2 variables that have been selected by the user to be graphed (current speed and y position). All other variables in the world model are accessible through the lower right window, where logging, charting, minimum, maximum, and other values are displayable. The lower left window displays the state table for one of the controller nodes (in this case, the Prim) for the vehicle, with the current state highlighted.

6 Summary

This paper has presented an integrated simulation framework that is capable of aiding researchers in developing, debugging, and evaluating algorithms, subsystems, systems, and groups of systems. While the entire framework has not yet been implemented, a prototype system does exist. This system allows the seamless operation of both real and simulated systems in an environment that contains both real and virtual features. In addition, standard interfaces, data logging facilities and ground truth exist to aid in the evaluation of the performance of systems and the comparison of multiple systems under repeatable conditions.

While interfaces exist for the "section" and "vehicle" level of the 4D/RCS architecture, it is desired to have interfaces that allow the testing and evaluation of components that reside at any architecual level. It is also desirable to be able to simulate additional environmental models for both the high and low fidelity simulators. These include traffic signals and signs for the low-fidelity simulator and a more realistic mobility platform simulator for the high-fidelity simulator.

Therefore, future work on this framework includes the development of additional interfaces, further development of the simulation environment, and the incorporation of the design and debug toolsets.

References

[1] J. Albus. 4-D/RCS reference model architecture for unmanned ground vehicles. In G Gerhart, R Gunderson, and C Shoemaker, editors, *Proceedings of the SPIE AeroSense Session on Unmanned Ground Vehicle Technology*, volume 3693, pages 11–20, Orlando, FL, April 1999.

[2] H. Huang, E. Messina, H. Scott, J. Albus, F. Proctor, and W. Shackleford. Open system architecture for real-time control using an uml-based approach. In *Proceedings of the 1st ICSE Workshop on Describing Software Architecture with UML*, 2001.

[3] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. RoboCup: A challenge problem for AI. *AI Magazine*, 18(1):87–101, 1997.

[4] E. Messina, C. Dabrowski, H. Huang, and J. Horst. Representation of the rcs reference model architecture using an architectural description language. In *Lecture Notes in Computer Science EUROCAST 99*, volume 1798 of *Lecture Notes in Computer Science*. Springer Verlag, 1999.

[5] K. Murphy, M. Abrams, S. Balakirsky, T. Chang, A. Lacaze, and S. Legowik. Intelligent control for off-road driving. In *First International NAISO Congress on Autonomous Intelligent Systems*, 2002.

[6] W. P. Shackleford, F. M. Proctor, and J. L. Michaloski. The neutral message language: A model and method for message passing in heterogeneous environments. In *Proceedings of the 2000 World Automation Conference*, June 2000.

[7] C. M. Shoemaker and J. A. Borenstein. Overview of the demo III UGV program. In *Proc. Of the SPIE Robotic and Semi-Robotic Ground Vehicle Technology*, volume 3366, pages 202–211, 1998.

[8] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A second-generation museum tour-guide robot. In *Proceedings 1999 IEEE International Conference on Robotics and Automation*, volume 3, pages 1999–2005, May 1999.

Evaluating the Performance of a Vehicle Pose Measurement System

Harry Scott
Sandor Szabo
National Institute of Standards and Technology

Abstract
A method is presented for evaluating the performance of a vehicle pose measurement system (e.g., GPS, inertial sensors, etc.). The method supports evaluations of the system on a vehicle moving at high speeds. An example is provided to illustrate the method.

Keywords
Vehicle, pose, position, measurement, accuracy, uncertainty, GPS, IMU

1 Introduction

Precise knowledge of the pose of a moving ground vehicle is important in such applications as navigation, safety, robotics, metrology and surveying. Vehicle pose measurement (VPM) system manufacturers typically state the accuracy of their systems under static conditions, that is, with the system stationary over time. Latency and synchronization errors are difficult to detect under static conditions. In this paper, a method is described for evaluating the performance of a VPM's ability to measure the position of a moving vehicle. The method requires that the VPM have a synchronization interface, which we call a sync pulse interface. This interface is often incorporated in a VPM to support photogrammetry applications where a camera's vertical sync latches a pose measurement. In the method presented in this paper, a photo emitter/detector sensor that senses energy reflected off a reflective target generates a sync pulse. By mounting the photo sensor on the vehicle and placing a reflective target on the ground, pulses are generated whenever the vehicle drives directly over the target. Repeated measurements of the target as the vehicle drives by are used to determine the repeatability in the VPM's measurement of the target location.

The National Institute of Standards and Technology (NIST) Intelligent Systems Division is pursuing work in several areas of intelligent vehicles, including research in autonomous mobility and development of performance measurement techniques. Various testbed vehicles are used in support of several projects, and each has been, or will be, outfitted with a VPM system. These systems are used in at least two distinctive ways. For autonomous vehicles, the VPM is a principal component of a real-time navigation system, providing necessary information for the vehicle to move through the environment. A somewhat different need is met by the ability of the VPM system to capture appropriate information in real-time that can be combined with other sources of information after the fact to generate more precise pose information than is available in real-time. This meets a metrology need, providing improved accuracy to evaluate performance of the vehicle itself and its various sensors.

This paper describes a method to determine or confirm the performance of an implemented VPM on a testbed vehicle. Since the VPM examined in this paper is intended for metrology applications, steps are included in the method to post process the data to obtain the highest possible accuracy. The method described could also be used to obtain real-time performance information. This is accomplished by analyzing the real-time solutions as opposed to the post-processed solutions. The method consists of a way to precisely trigger vehicle position measurements, a procedure for collecting data and a way to analyze the repeatability in the vehicle position measurements. Although VPMs are capable of providing a full position and orientation solution, this current effort addresses only the position measurement capability. In the following sections, the method is described in the context of evaluating a specific VPM, though the described method is applicable to any VPM that supports a trigger mechanism.

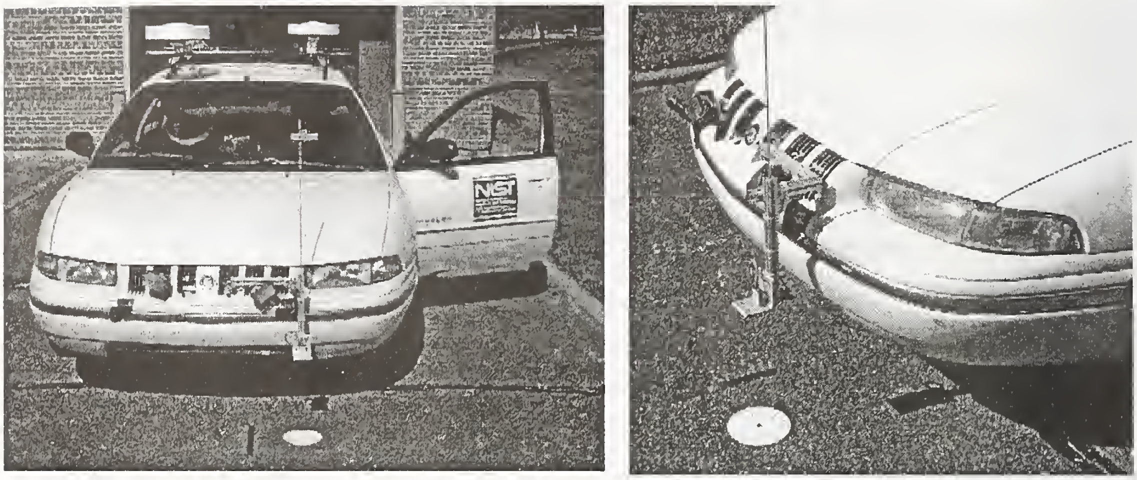


Figure 1 Note the downward looking photo sensor, the retroreflector on the ground and the vertical rod to guide the driver.

2 Test Procedures

The vehicle used in the evaluation is a full size passenger sedan. The installed VPM is a system that integrates a dual frequency carrier phase Global Positioning System (GPS), a secondary GPS, an inertial measurement unit (IMU) consisting of accelerometers and fiber optic gyros, a wheel encoder, and a system control and data collection computer. This system is configured to capture and provide solutions at a 200 Hz data rate. In addition, it is configured to capture all data required for later post processing.

The system is augmented with an external photo sensor connected to the sync pulse interface. When triggered, the result is the notation in a file of an event occurrence, along with the time of the event with microsecond time precision. Since triggers will be caused during vehicle motion, latency in the assertion of this signal will be reflected in position errors in the resulting data. For this reason, a low latency photo sensor (300 ms) was employed. During post processing, pose solutions are determined for selected events of interest. The sensor was mounted slightly in front of the vehicle bumper, in line with the vehicle driver position. A vertical rod above the sensor assists the driver in steering the sensor over a target while driving. The target is a simple retroreflective surface, constructed by affixing a layer of retroreflector material (available in sheets) to an aluminum disk. A disk of 15.24 cm (6 in) diameter is used. This size was selected as large enough to enable the driver to successfully steer the vehicle over it at speeds of interest most of the time, even in a grassy and somewhat bumpy field, but small enough to keep the target detection points close to a surveyed point. In this test a National Geodetic Survey (NGS) survey marker, flush with the ground, is used. The retroreflector is simply centered on top of it.

The configuration of the photo sensor, its mounting, and target is shown in Figure 1.

To evaluate the VPM's measurement performance, the vehicle is driven over the target from a variety of directions while collecting all data necessary to compute, via post processing, the position of the sensor at the time of target crossing. In the analysis phase, the repeatability of the measurements of the target's radius and the location of the target with respect to the survey marker produce a measure of performance of the VPM system.

The position solutions are computed for the location of the sensor (actually for a point on the ground directly below the sensor). This requires a transformation of coordinate frames between the VPM system and the position of interest below the sensor. This transformation depends on knowledge of the translations and rotations between the two reference frames. In the case of this implementation, translation and rotation measurements between the IMU reference frame and the sensor reference frame are needed. These measurements are entered into the VPM system. Determination of the translation and rotation parameters is performed in two steps.

The first is a best-effort measurement of the (x, y, z) distances between the IMU and the point below the sensor. Performing these measurements is physically awkward because of the location of the IMU in the vehicle trunk and the sensor location at the front bumper. An attempt was made to mount the IMU to align with the major vehicle axes as well.

Since these measurements are difficult and somewhat prone to error, data is also collected during the test run which is used to calibrate the system by adjusting the translation and rotation parameters as required. Several events are triggered by driving the vehicle sensor over the target as slowly ("creep") as possible and as well centered laterally as possible. These creep events are collected for approaches to the target from four directions. The slow speed is intended to eliminate any significant data latency errors. Positions of events collected in this manner should conform to the edge of the target, and the calibration parameters are adjusted slightly to yield solutions consistent with the known target diameter. This "creep" data also helps detect any movement of the sensor that may have occurred between tests.

After collecting the initial creep calibration data, the vehicle was driven over the target at a number of speeds and from a number of directions. Speeds were limited to about 10.6 m/s (≈20 mph) due to the roughness of the grassy field where these tests were performed. (The sensor/retroreflector system has been tested successfully at highway speeds on a roadway as well.) Approximately 40 events and their associated data were collected in this way, including a second set of creep points after completion of the vehicle at-speed runs.

3 Post Processing

Since the purpose of this test was to determine the maximum accuracy possible from the VPM, post processing was conducted on the data. If the purpose of a test is to evaluate the real-time performance of a VPM, then this step would not be performed.

The data logged on the vehicle during the test is retrieved, and the real-time navigation solution, though not the subject of this paper, is examined to confirm the existence of good data depicting an appropriate vehicle trajectory. The raw data is then post processed to obtain the more precise solution for the events of interest.

To enhance the quality of the solution, differential GPS post processing is used. A detailed explanation of this processing is beyond the scope of this paper, but the approach essentially makes use of information collected from another nearby GPS receiver (base station) at an accurately known location to remove (during post processing) certain kinds of errors from the reported position of the rover (our vehicle).

For these tests, we used a National Geodetic Survey (NGS) Continuously Operating Reference Station (CORS) located in Gaithersburg, MD known as GAIT.
The NGS, an office of NOAA's National Ocean Service, coordinates a network of continuously operating reference stations (CORS) that provide GPS carrier phase and code range measurements in support of 3-dimensional positioning activities throughout the United States and its territories. (See http://www.ngs.noaa.gov/CORS/)

The reference station data for station GAIT is downloaded from the NGS for the appropriate period during which we performed the test, and used by post processing software to enhance the GPS solution. Further post processing is performed to integrate this GPS solution with the raw data from the VPM system sensors (IMU and wheel encoder). Smoothing algorithms are executed, and interpolation of navigation solutions for the recorded events is performed. The result is a file with a full navigation solution for each event. The solution information in this file is analyzed below.

4 Analysis

Two types of VPM measurement errors are estimated: target (i.e., reflector) location and target radius. The estimated location error is derived as the difference between the surveyed (static) location of the target and VPM measured (dynamic) location of the target. The estimated radius error is the difference between the known radius of the target and VPM measured radius of the target. The location error may be left uncalculated if survey data is not available. The following process is used to compute uncertainties.

1. Put event points in a convenient local coordinate system. First, the event points (VPM-measured coordinates of target's edge) are transformed from latitude and longitude coordinates into UTM coordinates (see velvet.tec.army.mil/access/milgov/fact_sheet/geotrans.html) so that errors may be expressed in meters. Second, if available, the known location of the target is subtracted from each event point. This places the event points in a coordinate system whose origin is the surveyed location of the target.

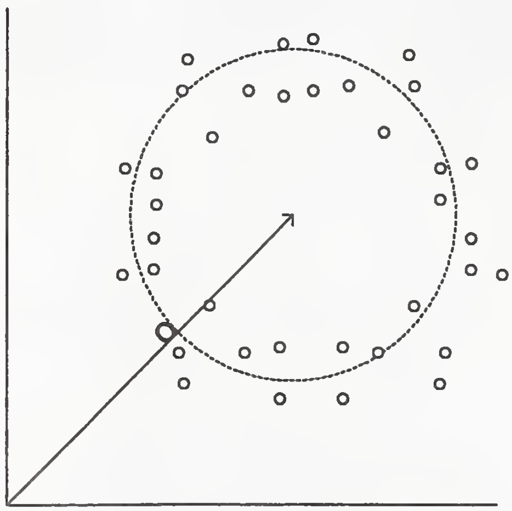


Figure 2. Example measurement points surrounding the target's edge.

2. Determine the target location error. A circle is fit to the event points. The fit produces an offset vector, **O**, indicating the center of the circle. Figure 2 shows an example set of data points with the offset vector to the center of the circle. If the origin of the coordinate system is the surveyed location of the target, then the magnitude of **O**, e_l, describes the estimated target location error.

3. Determine the target radius error and uncertainty. First, the offset vector of the circle is subtracted from each event point. This places the event points into a coordinate system with the center of the circle at the origin (see Figure 3). Then the distance of each event point from the origin, d_i, is computed. These distances are the measured radii of the target. The mean and standard deviation of the radius measurements, μ_r and S_r, are computed. The estimated target radius error, e_l, is the difference between the mean of the target radius measurements and the target radius measured by hand. Choosing a 95 % level of confidence, the component of the expanded uncertainty due to data scatter in the radius measurement is:

$$u_r = kS_r \quad (1)$$

Where $k = 2$ for $N \geq 30$ and k equal to a t-factor obtained from a t-distribution for $N < 30$.

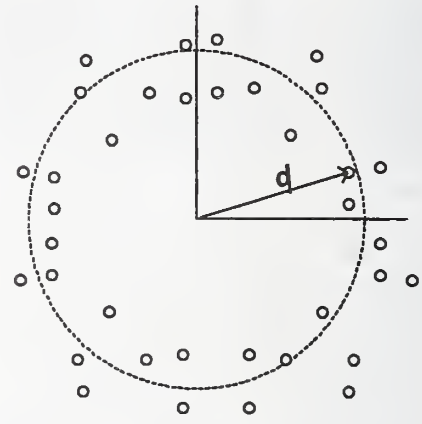


Figure 3. Measurements points translated such that center of circle is at origin. The distance of each point is the measured radius of the target

Figure 4 shows a plot of the event points collected when the vehicle traveled over a circular target with a radius of 7.62 cm (3 in). The origin of the plot coincides with the survey marker's coordinates, in this case, the coordinates of the NGS survey marker where the target was placed. The large circle is the target drawn to scale. Each event point is plotted as a small circle with an attached velocity vector (origin of vector is inside of small circle and points away from circle in direction of travel). The vectors are scaled to fit to the plot. The largest velocity vector is labeled 10.6 m/s to provide a scale for comparison (label is at end of vector opposite small circle shown on left side of plot). The results of the analysis of these measurements indicate a location error of 0.3 cm, an estimated radius error of 0.2 cm and a radius uncertainty of 1.3 cm (95 % level of confidence).

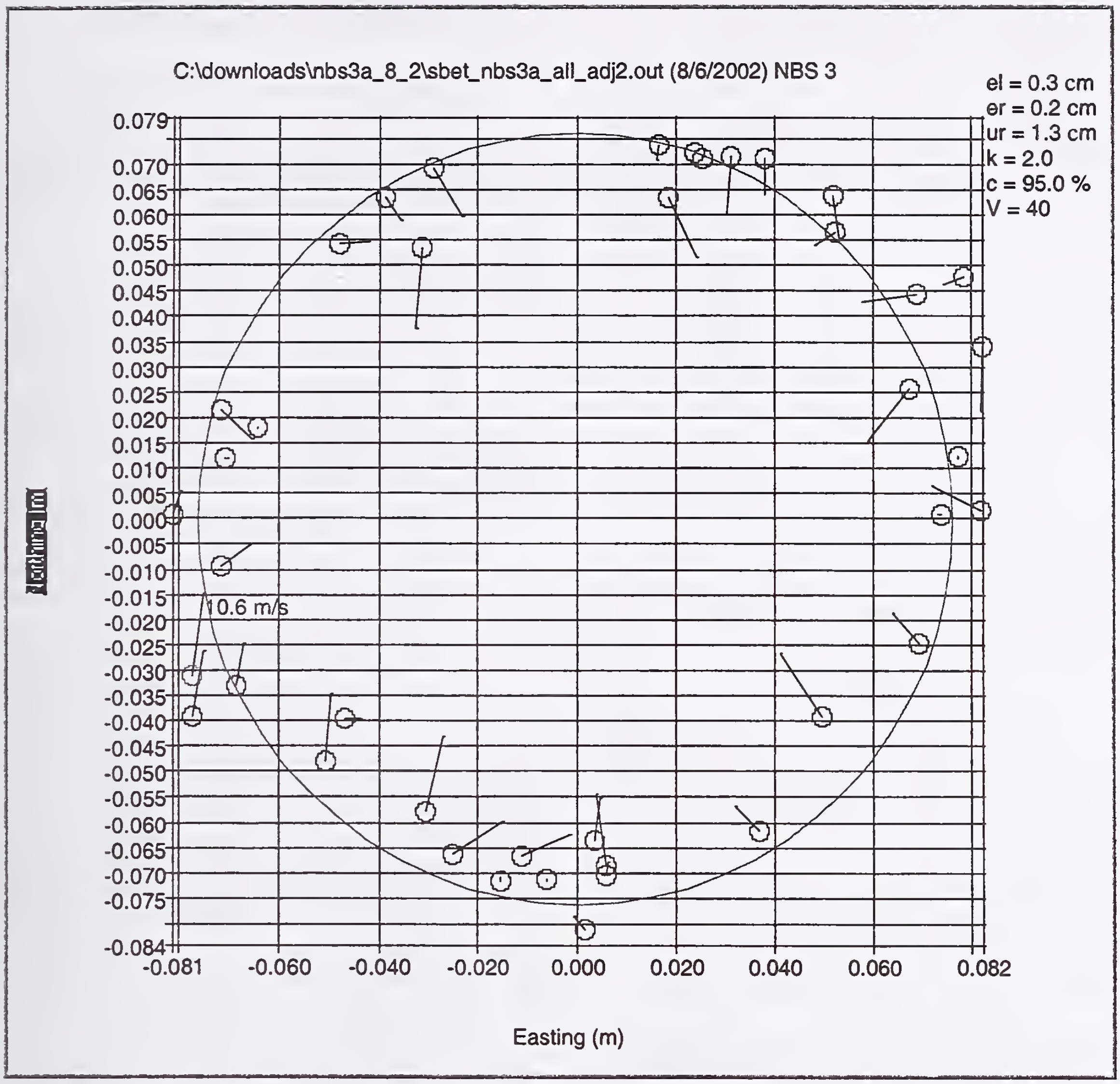


Figure 4 Plot of event points (shown as small circles) triggered when vehicle is driven over the edge of a circular target (large circle) with a radius of 7.62 cm. Vectors at each event point are vehicle velocity at event point. The largest velocity was 10.6 m/s (left side of target).

5 Conclusions

Further work is needed in several areas. This effort, an initial step in developing a performance measurement capability, focused only on vehicle position information in x and y. Height (z) data was collected but the data has not yet been analyzed. The testing methodology needs to be expanded to include orientation (roll, pitch, yaw) capabilities of VPM systems as well. That information, along with position information, is critical in registering the data received from intelligent vehicle sensors such as cameras and laser scanners, and directly affects the correctness of the model of the world being maintained by the vehicle computers. In addition, the scope of the work here included differential post processing of GPS data and further post processing of all the navigation sensor data. This is appropriate for metrology purposes. For navigation of intelligent vehicles, the real-time solution is important, and methods for determining the quality of the real-time navigation solution should be explored. Further, a real-time navigation solution may make use of real-time differential GPS corrections received by the vehicle during operation. The method described here can be used to examine position determination performance for those types of systems as well, and should be extended to characterize orientation measurement performance. Further, these tests were conducted with good GPS satellite coverage. While gaps in satellite coverage occurred, a test specifically excluding satellite info for prescribed periods would enable performance analysis of the VPM systems when they rely more on their inertial components.

We wish to acknowledge the National Highway Traffic and Safety Administration's Office of Vehicle Safety Research and the U.S. Army Research Laboratories' Robotics Program Office for their support of this work.

Performance evaluation of road detection and tracking algorithms

D. Dufourd and A. Dalgalarrondo
DGA/Centre Technique d'Arcueil
16bis, av. Prieur de la Côte d'Or, 94114 Arcueil Cedex, France,
Delphine.Dufourd@etca.fr, Andre.Dalgalarrondo@etca.fr

ABSTRACT

In this paper, we present a methodology to assess the results of image processing algorithms for unstructured road edges detection and tracking. We aim at performing a quantitative, comparative and repetitive evaluation of numerous algorithms in order to direct our future developments in navigation algorithms for military unmanned vehicles. The main scope of this paper is the constitution of this database and the definition of the assessment metrics.

KEYWORDS: *Image Processing Assessment, Outdoor Navigation, Ground Robotics.*

1 GOAL OF OUR WORK

In December 1999, the French defence procurement agency (Délégation Générale pour l'Armement) has launched a prospective program dedicated to ground robotics. Part of this program aims at developing autonomous functions for military unmanned vehicles navigation, such as autonomous road following, beacon and vehicle tracking and scene analysis. In this context, the Centre Technique d'Arcueil (CTA) of the DGA is currently conducting an evaluation of existing image processing detectors of unstructured road edges. The goal of this evaluation is to compare different road detection and following algorithms in a reproducible and quantitative way so as to direct future developments in navigation algorithms. It should allow us to determine the most promising techniques and possibly find orthogonal strengths between the algorithms so as to conceive hybrid and potentially more efficient methods. In this work, we plan to evaluate six road edges detectors coming from: Centre de Morphologie Mathématique (CMM) of the École des Mines de Paris [3], Laboratoire des Sciences et Matériaux pour l'Électronique et l'Automatique (LASMEA) [22, 1], Laboratoire Central des Ponts et Chaussées (LCPC) [7], the PG:ES company [23], and our laboratory [20].

The evaluation methodology is described in the following sections. Section 2 presents previous studies on performance evaluation. Section 3 focuses on our evaluation software environment named SENA. Section 4 describes the constitution of the image data base as well as the associated ground truth. Section 5 proposes different metrics to evaluate the algorithms with respect to the ground truth. Section 6 shows preliminary results concerning two road-following algorithms. Finally, section 7 concludes and outlines future developments.

2 EVALUATION METHODOLOGIES

2.1 *Assessment methods in image processing*

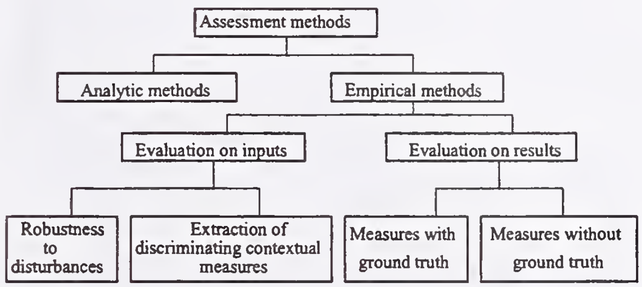


Figure 1: Classification of assessment methods.

In the last years, the image processing community has started to develop evaluation methods in order to be able to compare quantitatively the huge number of algorithms available after these last decades of research. Such an approach is very important for those who use image processing as a part of their research, like roboticists, since it provides a guide based on performance among the overwhelming available algorithms. However it should be noted that such an approach is very recent. For instance, Heath [12] has analyzed 21 papers on new contour detectors during the years 1993-96; the results are rather startling: while some papers do not even compare their method with other detectors, other papers use only 2 test images. Up to very recently, algorithms were not evaluated quantitatively, but only qualitatively on various criteria such as the neatness of their design or the sophistication of the underlying mathematical theoretical tools. Most experiments are conducted by human experts and lack any automation. The performance of the algorithm depends then on the know-how and the personal experience of the expert. Fortunately, the situation is changing, following the animated discussion of Jain and Binford [16], and there are always more special issues in journals or conferences focusing on image processing assessment issues.

Figure 1, taken from [6], shows a temptative general classification of methods for image processing assessment.

Analytic methods do not need an explicit implementation of the algorithm and take into account its general features such as its complexity, or the overall principles. Such methods can be used in the development phase when the designer has to choose which algorithms will be implemented on the robot. They allow a comparison of the algorithmic complexity and give an estimate of the time to be allotted to every algorithm, when the computing resources are known. The influence of the propagation of the variance of the input data on the results of the algorithm can also be estimated [11].

Empirical methods evaluate the algorithm by playing with its inputs and studying the evolution of its various outputs. The assessment of an algorithm can be done by varying the intrinsic parameters of the algorithm or by adding disturbances – noise, time-depending variation of the grey levels, saturation...– on the inputs and analyzing the evolution of the performance. Such an approach aims at defining the "satisfactory operating domain" of the algorithm. Such a knowledge is important in order not only to compare and choose the right algorithm but also to chain various algorithms, as it gives hints at the propagation of errors. A weak sensitivity to disturbances or modification of the tuning parameters is needed in an automatic system. Some methods use contextual hopefully discriminating measures in order to decide whether an input – in our case the current image – "suits" the algorithm, i.e. is in the "satisfactory operating domain". Measures that are correlated with the result of the algorithm are looked for.

Methods based on the measure of a difference between the results of an algorithm and a reference solution, called "ground truth", allow an automation of the assessment process. As shown in figure 2, the joint use of test images, ground truth and metrics, that yield a measure of the difference between the results and the ground truth, provides quantitative evaluation of the algorithms. Whereas the ground truth is generated by a human expert or by a reference algorithm, the variation of the tuning parameters of the algorithm follows predetermined ranges and sampling and can be fully automated, as well as the analysis of the results, as soon as the metrics have been explicitly given. This is the method we have selected for our assessment.

Finally, empirical evaluation methods without ground truth are based on the availability of empirical measures of what a "correct result" should be [6]. Such measures are built following intuition and/or successive experiments during the design phase, where ground truths may be used. Of course such measures are very dependent on the task to be performed by the algorithm but they can be automatized. For example, in [19], we present a robot control architecture which uses evaluation mechanisms in order to select automatically the most appropriate perception algorithm.

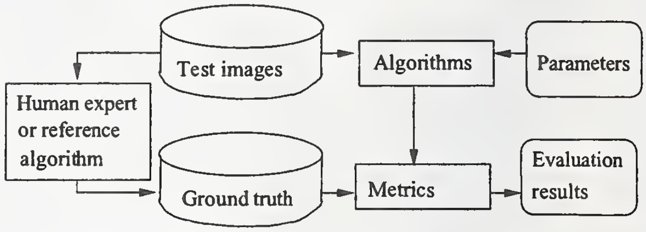


Figure 2: How to assess an algorithm when ground truth is available.

2.2 *Road following algorithms evaluation*

Although a wide variety of vision-based road following algorithms have been proposed and implemented over the last two decades, few techniques have been developed to assess their quality. Far too many articles rely on qualitative results, exhibiting a handful of example images to illustrate the performance of the algorithms while real applications would mean processing millions of images without making any serious error [17].

In many cases, the efficiency of road following algorithms is only characterized by the speed achieved by the whole autonomous system. For instance, in the field of autonomous lateral control on highways and marked roads, numerous experiments consist in driving a few thousands of kilometers and providing statistics about the performance of the system : maximum time elapsed between two manual interventions, average and maximum speed, distance between the vehicle and the lane, etc [5]. However, using such global characterizations, it seems difficult to determine exactly what makes the system efficient and what could be improved to make it better : is the autonomous vehicle fast because the road following algorithm has been implemented efficiently using powerful computationnal resources, because this image processing algorithm is very accurate and robust or because the control laws of the vehicle are well-designed?

Algorithms performing 3D reconstruction of the road have been evaluated in different ways. Guiducci performed indirect numerical tests on 1000 images, comparing the road width and vehicle speed estimated by his algorithm with their real values [10]. The actual road width was measured manually and the speed was given by the vehicle speedometer. However, these global test measures characterize the whole system, including the 3D road and vehicle models, while more direct measures would probably be helpful to improve the image processing algorithms more specifically. DeMenthon performed tests on both synthetic and real images [8]. Whereas the 3D profile of the synthetic data is known, the profile for the real data is reconstructed manually using a fusion between distance and video images. A specific task-oriented metric is used to assess the results of the algorithm: a reconstructed road is labelled "navigable" if the tracks of a two meter-wide vehicle following the centerline of the reconstructed road stay between the edges of the actual road over the whole reconstruction and do not cut these edges. However, manual 3D reconstruction is too time-consuming if the evaluation is to be performed on numerous data. Therefore, if a manual ground truth is to be used, it seems more realistic to operate directly in the 2D image space rather than in the real 3D world.

Finally, a few research studies focus on automating the measurement of ground truth for the evaluation of vision-based lane sensing. A NIST report on performance evaluation for robotic vehicles [13] proposed a specific device composed of a side-looking camera and a separate vision system to measure the offset between the vehicle and the lane. Using a detailed calibration of their imaging system and spectral measurement of the ambient illumination and scene, Everson et al. [9] generated images simulating various rates of precipitation. The metric used to evaluate their lane-sensing system consists of the variance lane centering behavior as a function of precipitation level. Kluge also performed a pilot study in order to get some insight into the issues involved in automatic performance evaluation of lane-sensing algorithms [17]. He selected a well-defined aspect of system performance in a single class of lane-sensing techniques. The ground truth was measured automatically using a reference algorithm and its correctness was hand checked on the 1800 windows of the data set. One can notice that automatic ground truth measurement requires a reference algorithm and possibly a specific equipment to measure the road edges, which is easier in the case of road marking detection than in the case of unstructured road edges detection with various environmental conditions.

3 THE SENA PLATFORM

Our laboratory is interested in various information and intelligence military systems which use image processing. Current researches address the evaluation of satellite image registration, infrared image segmentation, image fusion and interpretation. Applications of these algorithms on military systems must present specific qualities in order to cope with extreme battlefield situations. This leads to different system testings and notably to the development of a general evaluation architecture called SENA (Système pour l'EvaluatioN d'Algorithmes).

SENA is a customized software environment for fast algorithm implementation and evaluation of a wide range of applications. It helps in assembling image processing operators and replaying the experiments on a large amount of images. In a sequence of operators, tools for measuring or visualizing partial results can be incorporated. These tools can also be considered as image processing operators. Thus, SENA is able to organize and execute a sequence of operators of different types (source code, shell scripts, binaries, libraries) and origins (operators that were developed specifically or not for the platform). The only constraint is that all the operators must be executed on the same host computer. Practically, SENA runs on a SMP computer (SUN Enterprise 10,000 with 32 processors) to cope with huge amounts of data and important range variation of the algorithms parameters. SENA has been developed by Cril Ingenierie under CTA specification and supervision. Among other graphical software environments able to construct and execute sequences of image processing operators, Khoros is probably the best known. However, SENA is most likely the only platform allowing simultaneous use of various types of operators (scripts, binaries...), definition of cyclic graphs of operators, automatic parallel execution of the assessment process on range of data and parameters and coupling with a database.

4 DATABASE CONSTITUTION

The database includes the images that will compose the input of the image processing algorithms and the ground truth suited to the final task to assess. For our purpose, we need images of unstructured roads and trails taken from a ground vehicle whose size and mobility are close to the targeted UGV. Collecting these images is relatively easy and cheap with nowadays technologies. The two main difficulties are the representativity of the images, in relation with the missions and the environment of the UGV, and the constitution of the ground truth.

The first step is the specification of the hardware to grab images on the proving ground. This includes the vehicle, the camera (position, field of view, frame rate, resolution, type of sensor...), the grabbing device, the storage media and the image files coding. Specification of noise and saturation levels on the images and general ranges of climatic or illumination conditions can be added. If image calibration is needed by some algorithms, the acquisition of images of reference scenes must be specified. Moreover, data concerning the speed and the attitude of the vehicle can be attached to each image, in order to feed the environment or vehicle models which may be used by some algorithms.

The second step is the specification of film scripts for the image acquisition. In our case, we specify two kinds of scenarios: general ones with an increasing difficulty level for road edges extraction and special scenarios which are dedicated to road and trail particularities. In the first case, one gets homogeneous sequences of images in order to assess an algorithm all along a sequence without risking an irreparable failure on some images. In the second case, it is possible to evaluate the algoritm behavior in harsh conditions. The special scenarios must provide known difficulties for the algorithms like puddles, hairpin bend, abrupt road widening, slough, parked vehicles on the roadsides, changing soil, transversal and longitudinal road markings, etc. In practice, we defined six general scenarios and twelve special scenarios. The general scenarios belong to two categories: tarmac roads and gravel-mud roads. There are three scenarios for each category, with an increasing level of difficulty. Each scenario must correspond to a specific location on the proving ground in order to be recorded in about four different illumination and weather conditions. Some image sequences recorded at night with the vehicle lights and with a FLIR camera are also defined. The length of the image sequences may vary between 60 and 120 s which corresponds to a distance between 500 and 1000 m for a vehicle travelling at a mean speed of 30 km/h. As for the twelve special scenarios, the length of the image sequences is shorter (about 20 to 30s) in order to isolate each difficulty.

The image acquisition is currently being performed in DGA testing facilities situated near Angers. Figure 3 shows examples of images taken at this location. This first version of the database will count about 20,000 images of roads and trails. This amount accounts for the second main difficulty of the construction of this database. Indeed, on each image a human expert has to define the ground truth i.e. to draw the road edges on the images. For that particular task, we wrote a specification which contains rules to follow in order to decide where the road edges are in a given image. Then, in order to facilitate this long and dull job, we have created a program with a dedicated graphical interface which manages the name and numbering convention of the images and ground truth files of a sequence and allows, on a new image, an easy modification of the grountruth defined on the previous image.

5 EVALUATION METRICS

Hoover et al. [14] underlined the need for multiple metrics in image processing algorithms assessment, so that users can consider different aspects of the algorithms and choose the one which is best suited to their application. Following this point of view, we propose eight different metrics aiming at assessing geometrical accuracy as well as a good global correspondence between the ground truth and the output of the algorithms. As mentionned before, extracting 3D references from numerous data appears extremely time-consuming so that we have decided to work in the 2D image space. Therefore, our metrics are also computed in the 2D image space and do not consider the 3D real world data such as the width of the road or the pitch angle of the vehicle.

Among the various metrics available, we can distinguish contour-oriented metrics and region-oriented metrics, which reflect the dual approaches to image segmentation.

5.1 *Contour-oriented metrics*

Before computing most contour-oriented metrics, we need to perform a matching procedure between the reference road edges and the result of the algorithm. Indeed, we have to determine which parts of the extracted road edges correspond to given parts of the reference road eges. We chose the so-called "buffer method" described by Wiedmann et al. [25] in the context of automatic road axes extraction from aerial images. Using this technique, every portion of the extracted road boundary lying within a certain distance (i.e. the size of the buffer) from the reference boundary is considered as matched.

A survey realized in our lab by Capolunghi and Ropert [6] defines five different categories for common assessment measures, as listed below.

5.1.1 Measures of classification/detection errors

These measures consist in counting the number of pixels that have been misclassified by the algorithm and extracting detection and cover rates as well as statistical measures. Our first three metrics correspond to this category.

The **completeness metric** computes the difference between the length of a result judged as valid (within the buffer tolerance) and the length of the ground truth. It enables us to determine whether the algorithm has managed to find the whole road or only a small part of it. More formally, using the notations and configuration of Fig. 4, this metric is defined by:

$$m_1 = \frac{length(BC)}{length(AD)}, \quad m_1 \in [0,1]$$

The **correction metric** determines what portion of the result lies within the tolerance area. Using the notations and configuration of Fig. 4, it is defined by the following formula:

$$m_2 = \frac{length(GF)}{length(GE)}, \quad m_2 \in [0,1]$$

Finally, a **quality metric** combines the previous ones. The quality of a road edge estimated by the algorithm is regarded as good if the edge lies within the tolerance area and "explains" most of the reference edge. More precisely, this quality can be expressed as: $m_3 = m_1 \times m_2, \quad m_3 \in [0,1]$.

Wiedmann et al. defined similar metrics using notions of true positive, false positive and false negative for the output of the algorithms [25].

5.1.2 Measures of localization errors

Measures of localization errors compute a distance between two sets of points A and B (in the case of contours, one can consider that these sets are composed of the pixels that form the contour). Among them, we can mention the figure of merite proposed by Pratt [21], the Hausdorff distance and the Baddeley distance [2] . Huang and Dom [15] also proposed to evaluate the divergence between A and B by distance distribution signatures which correspond to distance histograms. Different statistics can be extracted

Figure 3: Examples of road images of the DGA testing facilities near Angers.

from these histograms such as the mean value and variance. We have opted for this last measure computing the average distance between the reference and result edges:

$$m_4 = \frac{\Sigma_G^E dist(algorithm, ground\, truth)}{length(GE)}, \quad m_4 \in [0, \infty[$$

Besides, we can compute other statistics concerning these distances such as variance, as well as maximal and minimal distances (which is akin to the Hausdorff distance).

5.1.3 Error classification

The evaluation of edge detectors is sometimes based on a classification of their errors. For example, the estimated edges can be labelled as well-detected contours, over- or under-segmented contours, missed contours and contours due to noise. Completeness and correctness illustrate some of these notions but we could also introduce new metrics involving "redundancy" [25] for instance. Fig. 4 illustrates this notion of redundancy: the length of the thick result edge far exceeds the length of the reference edge. However, the ground truth does not present many singularities and the algorithm results are usually smoothed by linear regressions or hyperbolic approximations, so that this redundancy metric potentially does not provide much information.

5.1.4 Parametric approach

The measure classes described so far are computed pixel by pixel from the output data. Conversely, the parametric approach consists in representing the data which we intend to compare by a few specific features. As a result, the output data are reduced to a single parameter vector. For instance, Strickland proposed linear combinations of local measures related to the shape of the contour (continuity, regularity and thickness), its location with respect to the ground truth and to contours due to noise [24]. The first three criteria are not well-adapted to our application since the contours provided by the evaluated algorithms are usually one-pixel thick, continuous and regular. The fourth criterion related to location is taken into account by the measures of localization errors described above. However, the last criterion is not used because we have not measured the noise levels in the images.

5.1.5 Non-scalar measures

Non-scalar measures can be linked to statistical approaches. For instance, Huang and Dom proposed distance histograms [15]. However, to avoid multiplying the measure data, we only keep the mean value for the histogram, and possibly the variance as well as the extreme values. Performance diagrams such as the Receiver Operating Characteristic (ROC) curves are often used to illustrate algorithms performance. We can draw similar curves representing $1 - m_2$ (which corresponds to a false positive rate) with respect to $1 - m_1$ (corresponding to a false negative rate) for different values of the algorithm parameters.

5.2 *Region-oriented metrics*

Contour-oriented metrics provide detailed information about the geometric accuracy of the algorithms. However, in sharp turns or on very irregular paths, pessimistic algorithms using a very simple road model (a triangle for example) risk being severely penalized by these metrics even if they find a drivable area within the boundaries of the real road. As a result, we have defined several metrics based on surfaces.

Whereas edge-oriented metrics need a preliminary matching process, region-based metrics can be applied directly since there is no ambiguity concerning their correspondence. However, in the general case, the road detectors provide open contours for the road, which means that we need to perform a closing procedure. We have decided to close the road region through linking the left and right upper ends as well as the left and right lower ends.

Region-oriented metrics can be divided into the same categories as contour metrics.

5.2.1 Measures of classification/detection errors

Among the metrics measuring frequencies of incorrect classification of pixels in the image, we can mention the Hamming distance [15] and the Vinet distance. However, such

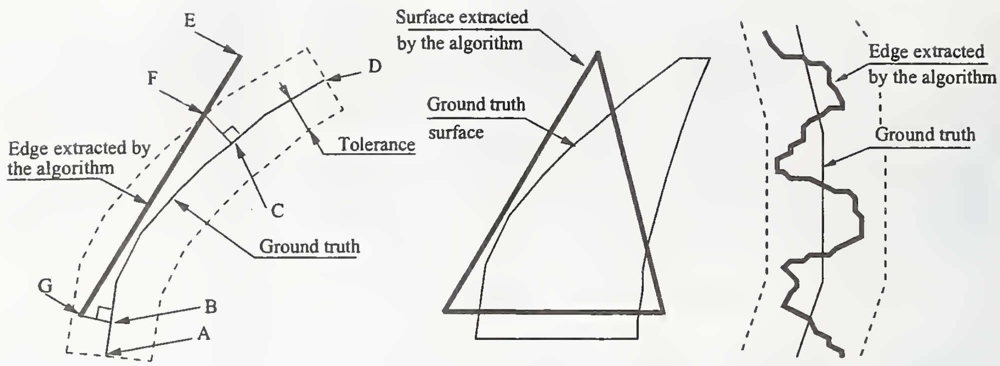


Figure 4: (left and center) Notations for metrics. (right) An example of redundancy.

metrics are designed to deal with region segmentation algorithms, and thus require a matching step between the result and the ground truth regions. Therefore, we can chose more simple measures (see Fig. 4 for the notations): a **completeness metric**:

$$m_5 = \frac{|S_{algorithm} \bigcap S_{ground\,truth}|}{|S_{ground\,truth}|}, \quad m_5 \in [0,1]$$

and a **correctness metric**:

$$m_6 = \frac{|S_{algorithm} \bigcap S_{ground\,truth}|}{|S_{algorithm}|}, \quad m_6 \in [0,1]$$

We can notice that combining m_5 and m_6, we can compute the Vinet distance. Besides, we can define m_7 as a combination of m_5 and m_6: $m_7 = m_5 \times m_6, \quad m_7 \in [0,1]$, and an overall **quality measure**: $m_8 = m_{3_{left}} \times m_{3_{right}} \times m_7, \quad m_8 \in [0,1]$.

5.2.2 Error classification

Hoover et al. proposed an error classification for extracted regions in the scope of image segmentation evaluation. They distinguished correct detection, over- and under-segmentation instances, missed detections and noise [14]. Once more, this classification is better adapted to multiple region matching rather than to a comparison between two regions. Nevertheless, we can notice that the basic values computed to perform this classification are based on boolean operations between pixel sets and correspond to combinations of m_5 and m_6.

5.2.3 Parametric approaches

Finally, concerning parametric approaches, various features of the regions can be computed and compared: surface, perimeter, moments, main axes, etc. Surface and perimeter are also taken into account in the previous measures while moments and main axes (or road axes) may provide interesting additional information.

6 PRELIMINARY EXPERIMENTATION

The image database has not been completly delivered and the algorithms are currently being integrated into SENA. We made a preliminary experiment concerning the metrics using two algorithms and one sequence of 224 images. Figure 5 shows the ground truth and the results of both algorithms on the same image. Figure 6 shows the values of the metrics along the image sequence.

Surface-based metrics (m_5 and m_6) appear far more stable than contour-oriented metrics, which is probably due to the severity of the "buffer method" for small values of the buffer width (12 pixels in our experiment, for 768 × 576 pixel images). The peaks in the diagrams indicate particular images for which the algorithms failed. For instance, the right edge determined by algorithm 1 on image 123 (see Fig. 5) presents poor values for m_1, m_2, m_4 and m_6. Algorithm 1 faces difficulties on images 81, 89, 90 and 103 as well (see m_1 and m_2), although m_4 indicates that these errors are minor compared to image 123. The end of the sequence presents a greater challenge for the algorithms since the vehicle arrives on a crossroad. As a result, the detectors tend to select a portion of the road which belongs to the intersection and which was not marked by the operator: completeness remains correct while correctness decreases. However, on the rest of the sequence, correctness is better than completeness, which means that part of the road is missed by the algorithms. The road detectors indeed have trouble finding the horizon line, so that the estimated boundaries do not extend to the upper part of the road. A metric that would only consider the lower part of the image would enable us to assess the quality of the algorithm whatever the estimation of the horizon line.

7 CONCLUSION

In this paper, we have described the complete methodology and various tools that will be used to assess the quality

Figure 5: Groundtruth (left) and results of algoritm 1 (center) and 2 (right) on image 123.

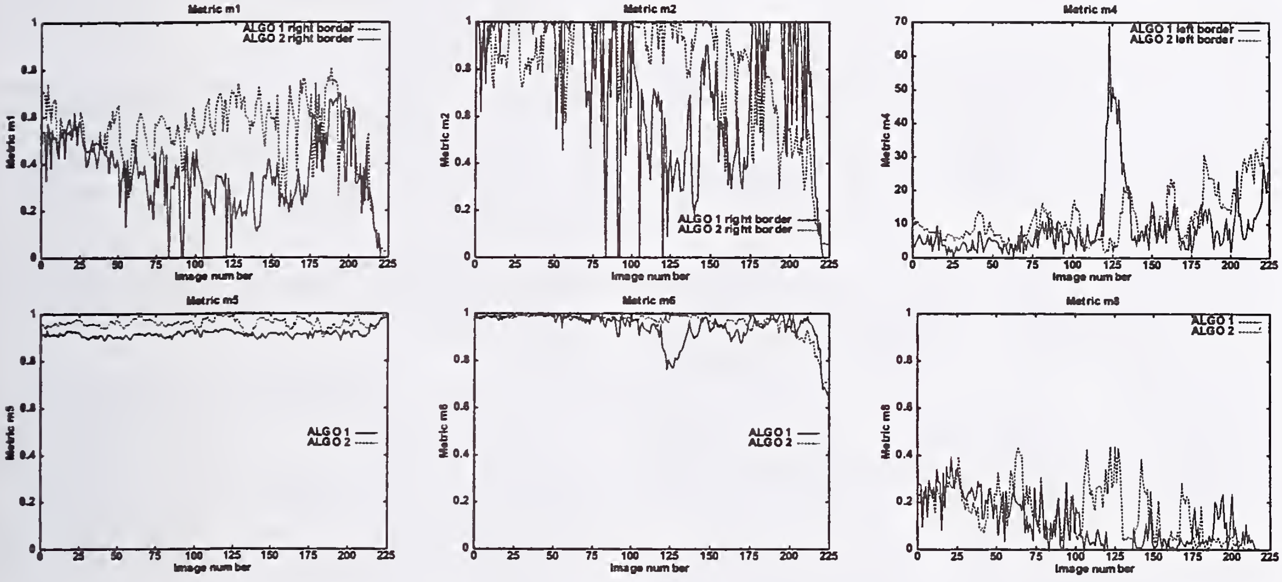


Figure 6: Examples of measures.

of unstructured road edges extraction algorithms. Within the next months, the image database should be completed and all the algorithms will be integrated into the SENA platform. This will allow us to apply our methodology to the whole data and compare the different edge detection techniques. Henceforth, this work offers many perspectives:

- Besides road edge detectors, we plan to apply our methodology to the evaluation of other vision-based algorithms which aim at enhancing the navigation capabilities of autonomous ground vehicles. Among them, we have selected beacon and vehicle tracking as well as image segmentation. The algorithms which we plan to test belong to three French laboratories: Laboratoire des Sciences et Matériaux pour l'Electronique et l'Automatique (LASMEA), Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS) and our laboratory.

- So far, we have defined six different metrics for the automatic assessment of edge detectors. However, we may come to modify these metrics if it turns out that they do not account for some qualitative phenomena observed by the operator during the evaluation. Indeed, Ropert and Capolunghi underline the necessity of a good correlation between the human judgement and the behavior of the metric [6].

- To go further, we could even use a specific methodology for choosing the metrics. Ropert et al. proposed such a methodology in the practical case of default detection in gammagraphy images of welded metal plates [4]. Letournel described a more sophisticated protocole in the field of aerial images interpretation [18]. She performed a statistical analysis in order to detect a relationship between objective metrics (given by mathematical formulas) and subjective metrics given by a human judgement (manual markings). Such an analysis would definitely be worth trying in the scope of our project.

- Among other metrics that could be tested, we can imagine measures which would be more oriented towards the specific task to be performed by the vehicle, such as the metric described by DeMenthon [8].

- Finally, it seems interesting to introduce metrics that would allow us to characterize more accurately the difficulty of the test images (signal to noise ratio or more sophisticated metrics such as the ones proposed by Kluge [17]). Such metrics should help us to build a more representative video database for the evaluation.

References

[1] R. Aufrere, R. Chapuis, and F. Chausse. A fast and robust vision based road following algorithm. In *IEEE Int. Conf. on Intelligent Vehicles*, Oct. 2000.

[2] A. J. Baddeley. An error metric for binary images. In *Robust Computer Vision: Quality of Vision Algorithms*, pages 59–78. W. Forstner and S. Ruwiedel, Mar. 1992.

[3] S. Beucher and C.-H. Demarty. Color segmentation algorithm using an HLS transformation. In *Mathematical Morphology and its Applications to Image and Signal Processing, Proceedings ISMM'98*. Kluwer, June 1998.

[4] J. Blanc-Talon and V. Ropert. A software architecture for algorithmic performance evaluation of vision systems. In *Proc. of the 3rd Workshop on European Scientific and Industrial Collaboration*, June 2001.

[5] A. Broggi, M. Bertozzi, A. Fascioli, and G. Conte. *Automatic Vehicle Guidance: the experience of the ARGO autonomous vehicle*. World Scientific Publishing, 1999.

[6] R. Capolunghi and V. Ropert. L'évaluation empirique en traitement d'images. Rapport technique DGA/DCE/CTA 2000 R 022, March 2000.

[7] P. Charbonnier, P. Nicolle, Y. Guillard, and J. Charrier. Road boundaries detection using color saturation. In *IX European Signal Processing Conf.*, 1998.

[8] D. DeMenthon. Inverse perspective of a road by local image matches and global 3D optimization. Technical Report CAR-TR-210, Univ. of Maryland, 1986.

[9] J. Everson, E. Kopala, L. Lazofson, H. Choe, and D. Pomerleau. Sensor performance and weather effects modeling for intelligent transportation systems applications. In *Intelligent Vehicle Highway Systems (Proc. SPIE vol. 2344)*, pages 118–128, 1994.

[10] A. Guiducci. Parametric model of the perspective projection of a road with applications to lane keeping and 3D road reconstruction. *Computer Vision and Image Understanding*, 73(3), Mar. 1999.

[11] R.M. Haralick. Propagating covariances in computer vision. In *Int. Conf. on Pattern Recognition*, Sept. 1994.

[12] M. D. Heath. A robust visual method for assessing the relative performance of edge detection algorithms. Master thesis, Univ. of South Florida, Dec. 1996.

[13] M. Herman, S. Szabo, K. Murphy, D. Coombs, T.-H. Hong, H. Scott, N. Dagalkis, K. Goodwin, and J. Albus. Recomandations for performance evaluation of unmanned ground vehicle technologies. Technical Report NISTIR-5244, NIST, Aug. 1993.

[14] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.

[15] Q. Huang and B. Dom. Quantitative methods of evaluating image segmentation. In *ICIP'95*, 1995.

[16] T. Jain and R. Binford. Ignorance, myopa and naiveté in computer vision systems. *CVGIP: Image Understanding*, 53:112–117, 1991.

[17] K. C. Kluge. Performance evaluation of vision-based lane sensing: some preliminary tools, metrics and results. In *IEEE Conf. on Intelligent Transportation Systems*, 1997.

[18] V. Letournel. *Evaluation d algorithmes d interprétation pour le renseignement d'origine image.* PhD thesis, DGA/Centre Technique d'Arcueil, 2002.

[19] D. Luzeaux and A. Dalgalarrondo. Assessment of image processing algorithms as the keystone of autonomous robot control architectures. In *Imaging and Vision Systems: Theory, Assessment and Applications*. NOVA Science Books, 2001.

[20] G. Martinoty. Suivi de routes et détection d'intersections. Rapport de stage de DEA IARFA, Univ. Paris VII, Aug. 2001.

[21] W. K. Pratt. *Digital Image Processing*. Wiley, John & Sons, New-York, 1978.

[22] C. Rouquet, F. Chausse, R. Chapuis, and P. Bonton. Segmentation non supervisée d'images de scènes routières : une approche multi-critères. *Traitement du Signal*, 13:197–208, 1996.

[23] F. Sandt and T. Skordas. Color vision for road detection system. In *Session LVIII - Progress in Picture Processing*. NATO Advanced Study Institute, Aug.-Sept. 1992.

[24] R. N. Strickland and D. K. Cheng. Adaptable edge quality metrics. *Optical Engineering*, 32, 1993.

[25] C. Wiedmann, C. Heipke, H. Mayer, and O. Jamet. Empirical evaluation of automatically extracted road axes. In *Empirical Evaluation Techniques in Computer Vision*. IEEE Computer Society Press, 1998.

A Platform for Studying Locomotion Systems: Modular Reconfigurable Robots

Ying Zhang, Craig Eldershaw, Mark Yim, Kimon Roufas, Dave Duff
System and Practice Lab, Palo Alto Research Center
Palo Alto, CA 94304
Email: yzhang,celdersh,yim,kroufas,dduff@parc.com

ABSTRACT

There are many fundamentally different mechanical motions that a system can use to achieve locomotion. Two standard examples are the wheels on a car or the legs of an artificial ant, but many others exist as well. As with all systems, there is an obvious desire to quantify how "well" each locomotion method performs. Unfortunately, as with many metrics, this is far from being a well-defined problem. Apart from the usual difficulty of deciding exactly what is the most important measure (peak speed, efficiency, etc), there is the question of divorcing the underlying locomotive concept from the particular implementation (just like a universal machine such as Turing Machine divorces hardware implementations from algorithms).

This paper proposes a particular platform which the authors believe can be used as part of a standard system for evaluating many different means of locomotion. Since one of the fundamentally different aspects of each of these locomotive methods is the underlying mechanism, then any standard platform must be capable of changing its shape and fashion of moving so as to be able to faithfully perform the locomotion to be tested. The PolyBot system, developed at PARC, is capable of just this.

KEYWORDS: *locomotion systems, locomotion gaits, modular self-reconfigurable robots, experimental platforms, performance measures*

1. MOTIVATION

Locomotion is an important attribute for many intelligent systems. All known intelligent species of life are capable of locomotion by some means or other. The focus of this paper will be a little narrower, focusing only on locomotion on solid surfaces (thus excluding swimming or flying). Over the last century human beings have invented various kinds of locomotion systems for motion over ground, mostly for fast and efficient transportation. Probably the two most widespread of these are as cars and trains. Most cars or trains cannot be considered intelligent systems, because (1) they are not autonomous to any significant degree, (2) despite a large internal sensor network, their perception of the outside world is very limited, (3) they are intended for use in very specialized artificial environments -- (cars on highways and trains on railways).

The world is being constantly changed through the increasing availability of progressively cheaper and more powerful computation. Predictions have been made suggesting that in twenty years time, cars and trains will become intelligent robotic systems. Like animals, these vehicles will not only have a brain (central control) but also nervous systems (networking) connecting all sensing and actuation components.

The majority of existing man-made locomotion systems is wheeled, since that is simple and efficient in a conveniently engineered environment (flat surfaces or rails). However natural locomotive systems (such as used by animals) have almost exclusively favored employing legs. The use of wheeled vehicles is largely limited to flat environments. Tracked vehicles tend to handle a wider variety of terrain but suffer in efficiency. Legged machines tend to be less efficient and harder to control but have the potential of traversing an even wider variety of terrains. While much research has been done on legged locomotion, little has been usefully commercialized. Even though legged locomotion is generally recognized to be more flexible, and has the potential to effectively traverse natural environments, as yet more knowledge and understanding of how to engineer such systems is needed.

It is hard to compare two locomotion systems with radically different design, or two systems engineered for use in different environments. This paper proposes the use of modular self-reconfigurable robots as a standard platform for studying various types of locomotion and developing concrete performance metrics. By using this one platform for testing all locomotive ideas, the fundamental locomotive principle being tested is somewhat divorced from the specific physical implementation.

A modular self-reconfigurable robot, named PolyBot, has been developed over the last three years at the Palo Alto Research Center (http://www.parc.com/modrobots). PolyBot consists of many component modules (possibly hundreds), each of which has sensing, actuation and computation. These modules can be configured into many different shapes, such as wheels/loops, snakes and centipedes. It is due to this versatility that PolyBot is able to implement a wide variety of different locomotive systems, allowing concrete performance metrics to be calculated and clear comparisons to be performed.

With PolyBot, it is possible to develop various types of locomotion gaits for different types of configurations, and study the effectiveness of various control strategies. The results can be used to develop the performance metrics, which in turn allows quantitative improvements to be made in the quality of locomotion systems.

This paper is organized as follows. Section 2 characterizes some initial concepts on locomotion systems and gaits, Section 3 discusses terrain evaluations, Section 4 presents locomotion performance metrics; Section 5 describes more completely PolyBot, the modular reconfigurable robot. Finally, there are possible directions for future research using PolyBot as a platform for studying locomotion systems.

2. LOCOMOTION SYSTEMS AND GAITS

A *locomotion system* is a powered system being able to move from one position/orientation to another. There is a considerable body of knowledge on animal locomotion [1] and vehicle locomotion [2]. The most typical classification of land locomotion divides locomotion into four types: wheeled, tracked, legged, and other. From authors' point of view, this is unsatisfactory for several reasons. First, the last area is a catchall, and would include such dissimilar means of locomotion as snake-like sidewinding, concertina, screw locomotion, etc. Second, there are too many instances of ambiguity. For example, a child cartwheeling may be considered legged locomotion since the child has legs. Would a spoked wheel with no rim or partial rims also be considered legged locomotion? Tracked locomotion is defined as traveling on endless belts. Is a belt around a tire then tracked locomotion? What about a slightly flat tire? Yim [3] in his PhD thesis in 1994 studied various locomotion systems and first characterized locomotion gaits systematically.

A *locomotion gait* is defined as one cycle of a pattern of motion that is used to achieve locomotion. There are *simple* gaits and *compound* gaits; compound gaits are combinations of two or more simple gaits. There are maybe finite classes of simple gaits, but combination of these can generate infinite number of compound gaits. For example as wheeled locomotion is one type of locomotion and bipedal walking clearly is another, the two can be combined as with a person wearing roller skates.

A large portion of ground-based locomotion gaits can be characterized as *statically stable gaits*. To achieve statically stable locomotion in general, one has to repeatedly do three things in any order:

1. remove ground contact points from the rear,
2. place ground contact points in front,
3. shift weight forward.

Throughout all of these steps, maintain static equilibrium throughout all motions. A statically stable gait defines a cyclical pattern that achieves these steps.

The simple ground-based statically stable locomotion gaits are characterized by three categories [3]: *(R)oll/(S)wing, (D)iscrete/(C)ontinuous, (B)ig/(L)ittle Footed.* For examples, a 4-wheel passenger car is RCL, a treaded tank is RCB, a cockroach is SDL, an earthworm is SCB, human is SDB, etc.

Yim [3] also characterized three fundamental ways that a simple gait may be combined: *articulated*, *hierarchical* and *morphological.* Articulated combination is to unite more than one locomotion systems, e.g., track and trailer. Hierarchical combination is to add one locomotion system on the top of another, e.g., roller skating. Morphological combination is to merge locomotion systems with different axis, e.g., a rolling sphere.

When deciding which gait would be most appropriate for a given situation, it would be useful to know the characteristics of each type of classification. For simple gaits, rolling systems tend to be simpler and more efficient. Continuous motion can be smoother over hard flat terrains. The larger the footprint, the better the performance in terms of speed, efficiency and mobility, etc. For compound gaits, single chain articulated gaits have several desirable features: the ability to travel in highly constrained areas, to fit between or cross large obstacles, with a large payload. Hierarchical gaits can achieve higher speeds than individual gaits, e.g., walking on a moving track belt is faster than walking on a ground. Morphological gaits add degrees of freedom to locomotion, which make the system more flexible.

3. TERRAIN EVALUATIONS

Simply comparing the locomotion capabilities of a horse to a wheeled car is meaningless, just like comparing apples and oranges. In nature, each form of locomotion exists in the environment that fits it best. Locomotion performance metrics will not be complete without terrain evaluations. Yim [3] defined the taxonomy of terrain effects (Figure 1).

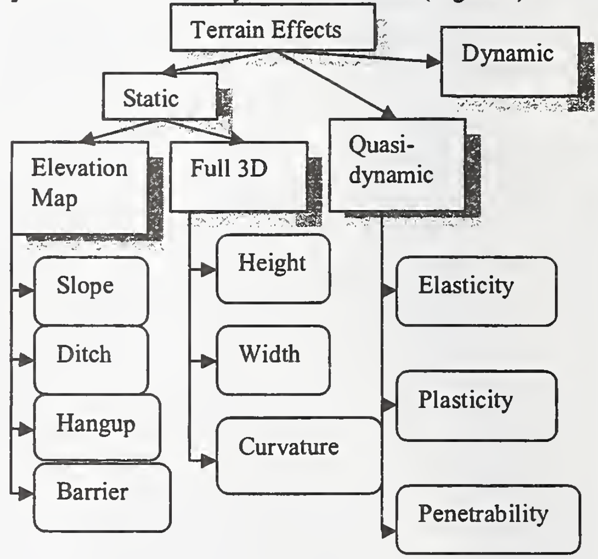


Figure 1: Taxonomy of Terrain Features

Static 2.5D terrain features include *slopes*, gradual elevation with height, *ditches*, holes in the ground, *hang-ups*, bumps in the ground, and *barriers*, a vertical object to cross like a wall. Full 3D terrain features include *height* constraints, obstacles on the top, *width* constraint, obstacles on the sides, and *curvature* constraint, the radius to turn. Quasi-dynamic terrain features include *elasticity*, *plasticity* and *penetrability* of ground surface, for example, a locomotion system will perform differently on soft mud terrain and hard wood floors. Dynamic terrain features include moving wind/current, moving terrain and obstacles, etc.

4. LOCOMOTION PERFORMANCE METRICS

The complexity inherent in intelligent systems means that it is rare for a useful metric to measure just one aspect of performance. In general, the evaluation function which serves as the metric will have multiple objectives which must be appropriately scaled and weighted. There are two main types of measurements for locomotion systems: system centric and environment centric.

In system-centric measurements, the type of environment is fixed (e.g. a dirt road) and other aspects (e.g. *speed*, *acceleration*, *efficiency*, *stability*, *payload*) are compared. In an environment-centric measurement, the value is some measure of the terrain which can be traversed (e.g. how steep the environment can be, or how rough) in terms of metrics of slopes, ditches, hang-ups, barriers, height, width and curvature constraints etc.

In addition to mechanical capabilities, computational capabilities can also be measured, such as *adaptability*, *robustness*, *self-repair-ability*, as well as the underlying computational components such as *CPU speed*, *memory*, *communication rate*, etc.

In addition to these "hard" measurements of locomotion systems, there are also "soft" measurements reflecting quality of system design (both hardware and software); these include *maintainability*, *modularity*, *scalability* and *reusability*. Some of these attributes are not directly related to performance, but are relevant to the total cost of ownership. Others are even less of interest to an end-user, but are still of importance for decision-making by the developer.

The authors do not claim to have yet developed a complete metric which satisfies all the conditions discussed above–this is work in progress. However they do put forward the idea of employing a uniform platform for locomotion testing. This, they argue, will simplify the measurement and comparison process, allowing effort to be directed towards refining the base metric. This universal platform is described in the next section.

5. MODULAR ROBOTICS PLATFORM

A modular reconfigurable robot is one that consists largely (or entirely) of identical components which can be assembled into many topological configurations. These different configurations generally equate to different physical shapes – each with different abilities and limitations. In this way the platform can be used to test many fundamentally different forms of locomotion. The platform proposed here is a modular *self* re-configurable robot: one that can change from one configuration to another autonomously. While this capability is not actually essential for its use as a universal platform as proposed here, the sensing, distributed computation, communications and control middleware required to support self reconfiguration will prove useful in carrying out the measurements for the performance metric. There are a growing number of modular self-reconfigurable robotic [4][5][6][7][8][9]. This paper focuses on one particular modular self-reconfigurable robot, named PolyBot [10].

PolyBot, is a modular reconfigurable robot system composed of two types of modules, one called a *segment* and the other called a *node*. The segment module has two connection ports and one degree of freedom (DOF) motion. The node module is a rigid cube with six connection ports but no internal DOF. PolyBot has been designed for applications including planetary exploration, undersea mining, search and rescue and other tasks in unstructured, unknown environments. PolyBot has been developed through its third generation at the Palo Alto Research Center. The latest design features smaller module size (5cm), more sensors (IR range, touch, force) and multiple actuators for locomotion, manipulation and reconfiguration, as well as bridged networks using CAN (Controller Area Networks).

Each PolyBot module has a Motorola PowerPC *MPC555* embedded processor with 448K internal flash ROM and 1M of external RAM. Software architecture has been developed for PolyBot, with a higher layer CAN protocol MDCN (Massively Distributed Control Nets) [11][12] and an Attribute/Service Model [11][13] for coordination of multiple tasks in multiple processes.

PolyBot is a good platform for studying various forms of locomotion. The PolyBot systems have demonstrated versatility by showing multiple modes of locomotion with a variety of characteristics, distributed manipulation and the ability to self-reconfigure. PolyBot can be configured into various shapes (see Figure 2,3,4). Each configuration has pros and cons in terms of performance. Snakes can traverse terrains with narrow entrance, such as pipes, and is the most robust among other configurations. Loops or wheels are most efficient over flat terrains. With deformed loops (conformance to terrains) it can also traverse effectively over stairs. Centipedes or spiders are good for avoiding obstacles and traverse rough terrains.

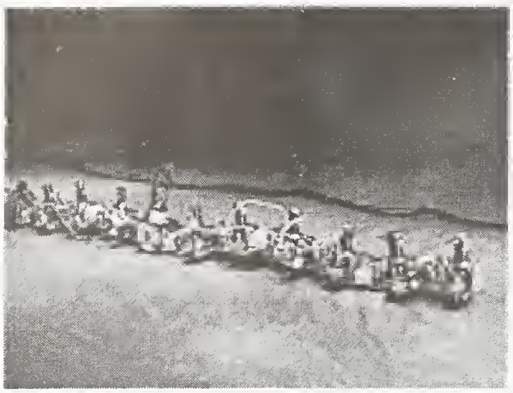


Figure 2: Snake Configuration

Figure 3: Loop Configuration

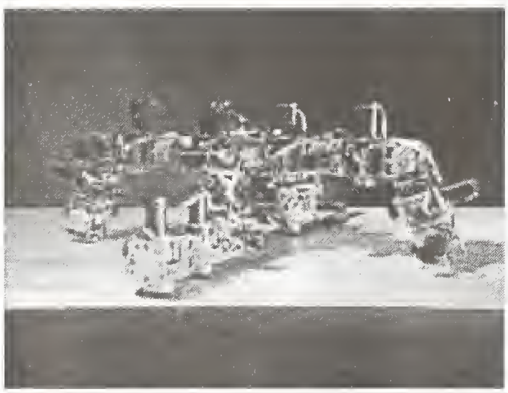


Figure 4: Spider Configuration

6. FUTURE WORK

There will be 100+ PolyBot modules built by the October this year. Various locomotion configurations and gaits will be tested and compared in the near future. A more complete understanding of and development of locomotion performance metrics will commence.

ACKNOWLEDGEMENT

This work is funded in part by the Defense Advanced Research Project Agency (DARPA) contract # MDA972-98-C-0009.

REFERENCES

[1] Gray, J., "Animal Locomotion", W.W. Norton & Co. Inc., New York, 1968

[2] Bekkar, M. G., "Introduction to Terrain Vechicle Systems", Ann Arbor MI, The University of Michigan Press, 1969

[3] Yim, M., "Locomotion with a Unit-Modular Reconfigurable Robot", PhD Thesis, Stanford University, 1994

[4] Fukuda, T. and S. Nakagawa, "Dynamically Reconfigurable Robotic System," Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 1581-1586, 1988.

[5] Kotay, K., D. Rus, M. Vona, C. McGray, "The Self-reconfiguring Robotic Molecule," Proc. of the IEEE International Conf. on Robotics and Automation, pp424-431, May 1998.

[6] Murata, S., H. Kurokawa, S. Kokaji, "Self-Assembling Machine," Proc. of the IEEE International Conf. on Robotics and Automation, pp441-448, May 1994.

[7] Pamecha, A., C. Chiang, D. Stein, G.S. Chirikjian, "Design and Implementation of Metamorphic Robots," Proc. of the 1996 ASME Design Engineering Technical Conf. and Computers in Engineering Conf., Irvine, California, August 1996.

[8] Unsal, C. and P.K. Khosla, "Solutions for 3-D Self-reconfiguration in a Modular Robotic System: Implementation and Path Planning," Proc. of SPIE Sensor Fusion and Decentralized Control in Robotic Systems III, Vol. 4196.

[9] Will P., A. Castano, W-M Shen, "Robot modularity for self-reconfiguration," SPIE Intl. Symposium on Intelligent Sys. and Advanced Manufacturing, Proceeding Vol. 3839, pp. 236-245, Sept. 1999.

[10] Yim, M., D. Duff, K. Roufas, "PolyBot: a Modular Reconfigurable Robot" Proc. of the IEEE Int. Conf. on Robotics and Automation, April 2000.

[11] Zhang, Y., K. Roufas, M. Yim, "Software Architecture for Modular Self-Reconfigurable Robots", IEEE/RSJ International Conference on Intelligent Robots and Systems, Hawaii, 2001.

[12] Zhang, Y., K. Roufas, M. Yim, "Massively Distributed Control Nets for Modular Self-Reconfigurable Ro-bots", AAAI Spring Symposium for Intelligent Distrib-uted and Embedded Systems, 2002

[13] Zhang, Y., M. Yim, K. Roufas, C. Eldershaw, "Attrib-ute/Service Model: Design Patterns for Distributed Coordination of Sensors, Actuators and Tasks", sub-mitted to Workshop on Embedded Systems Codesign, 2002.

PART I
RESEARCH PAPERS

1P2 - Performance of Planning Systems

Performance of Planning Systems

A. Meystel
Drexel University, National Institute of Standards and Technology

Planning is learning from experience in the domain of imagination.
From the Section 10 of this paper

The goal of this paper is to help in the organization of further research in the area of Planning.

1. Emergence and Development of the Theoretical Domain on Planning

The area of planning is a victim of linguistics: professionals of different domains give different interpretation to the phenomenon of "planning." Traditionally, it was associated with human activities and the help of science was expected

a) in a better organization of information for planning supporting the way the humans plan

b) in proposing of techniques that help to come up with "interesting" alternatives of planning decisions

c) in structuring the process of planning so that to make it more efficient

d) in modeling human activities during the process of planning.

Simon and Newell were the first that visualized planning as an element of each problem solving process. However, AI treated planning in the way it treated other problems - with the help of toy-problems (like Hanoi-Tower and block-world situations.

Specialists in control did not realize and did not appreciate the fact that feedforward control is actually the result of planning. The elegant discoveries of Nilsson and Fikes in their heuristic search methods remained unnoticed by control community until recently.

Specialists in automation of therapeutic solutions did not realize that they plan the process of disease development and plan the process of healing.

Specialists in education did not realize that proposing a curriculum and/or a particular syllabus has no difference from the processes of feedforward control and process of disease development and healing.

Specialists in optimization (re: operation research) did not realize that their problems were just elements of the planning theory.

Specialists in cognitive psychology seldom saw anything in common between "planning" and "imagination."

Thinkers of finding the goal did not have anything in common with thinkers computing schedules of goal achievement.

A common wisdom about planning was (and is) that there are many ways of planning and each of them has its place.

The following linkages were totally neglected:

a) the linkage between off-line planning and on-line control,

b) the linkage between specifying the goal and finding a schedule of achieving the goal.

c) the linkage between the linguistics of finding the image of "goal" and searching of the best schedule.

d) the linkage between the methodology and the *performance of planning*.

2. Planning for Behavior Generation

Robotics became the integrated domain that provided for blending the goals and testing the means of achieving them, i.e. a domain with a direct need for planning. In 1983, T. Lozano-Perez has introduced the idea of search in "configurations space". From the experience of using this search, it became clear that the exhaustive search would be computationally prohibitive if the configuration space is tessellated with the accuracy required for motion control. But his theory made two important thing obvious: 1) planning is an apogee of *creating admissible alternatives* and searching for the trajectories entailed by these alternatives. 2) planning is performed upon *milestones in the state space*, and if they do not exist in the reality, they should be created artificially (re: centers of the tessellata in the configuration space).

This development helped to realize that planning should combine the exhaustive (or meaningfully thorough) search off-line, and an efficient algorithm of an off-line control. It was about of this time that we stopped talking about control of actions and introduced a more balanced term of Behavior Generation. The latter became a codeword for the joint process of arranging and testing the alternatives within the mechanism of "planning" (open loop, feedforward control) blended with the on-line finding the alternatives of feedback for error compensation (closed-loop control, or "execution").

Behavior Generation alludes to many mechanisms of planning and execution. At the present time, these mechanisms cannot be considered as known thoroughly, and the general theory of planning can hardly be immediately attempted. There is a merit in discussing a subset of problems in which the goal is determined as attainment of a particular state.

The following are scattered notes on the progress in the domain of planning.

- Most of the realistic problems can be translated into this paradigm. Other types of realizations can also be imagined: in chess the goal is clear (to win) but this goal cannot be achieved by achieving a particular position in a space (even in a descriptive space.) Most of the problems related to the theory of games and linked with pursuit and evasion are characterized by a similar predicament.
- Let us notice the following: no matter what is the domain of decision making, the process of planning can be performed only by searching the state space and thus, determining both the final goal, and the trajectory of motion leading to this goal.
- In 1981 J. Albus has introduced the methodology of task decomposition for hierarchical systems which has grown into a NIST-RCS methodology.
- In 1981 G. Giralt outlines the concept of planning for mobile robots via tessellated space.
- In 1983, T. Lozano-Perez has introduced search in Configuration Space.
- In 1986, A. Meystel has demonstrated (CDC, Athens) that the most efficient functioning of a multilevel learning/control systems can be provided by a proper choice of a ratio of lower level/higher level of resolution. This concept of planning/control becomes a strong theoretical support for the hierarchical architecture of intelligent system control developed by J. Albus during the period of 1980-1998.
- In 1985-87 M. Arbib's school of control via "schemata" came up with a numerous schemes of "reactive" behavior. This gave birth to a multiplicity of robot control concepts which explore and exercise reactive behavior generation.
- In the meantime, the primary focus of robotics shifts to the area of systems which do not require any planning (robotics with "situated behavior"). Thus, the interest in planning diminishes (R. Brooks, MIT, R. Arkin, Georgia Tech) and the curiosity of researchers shifts toward emerging phenomena in non-intelligent robots.

3. Planning in a Representation Space with a Goal

This is an outline of the common methodology of planning pertaining to most of the disciplines and areas of application. The *world* is assumed to be judged upon by using its State Space (or the Space of Representation) which is interpreted as a time tagged vector space with a number of important properties. Any activity (motion) in the World (Space of Representation) can be characterized by a trajectory of motion along which the "working point" or "present state" (PS) is traversing this space from one point (initial, or state, IS) to one or many other states (goal states, GS.) The goal states are given initially from the external source as a "goal region", or a "goal subspace" in which the goal state is not completely defined in a general case.

From the point of view of planning, state space does not differ from the configuration space. Indeed, the upcoming behavior is represented as a trajectory in the state-space (and/or configuration space). One of the stages of planning (often the initial one) is defining where exactly is the GS within the "goal region." In this paper, we will focus upon planning problems in which one or many GS remain unchanged through all period of their achievement. Traversing from IS to GS is associated with consuming time, or another commodity (cost). So, the straight-forward exhaustive search is feasible which allows for exploring all possible alternatives.

Researchers in the area of reactive behavior introduced a method of potential fields for producing comparatively sophisticated obstacle avoiding schemes of motion. Reactive behavior is considered to be an anti-thesis for planning. It is not so. Planning based motion can be called reactive, too. The difference is that in the papers on reactive behavior, we react to the present situation. In the system with planning, we react too: but we react to the anticipated future.

Thus, planning can be considered an *anticipatory reactive behavior.* The difference is in the fact that anticipation requires representation richer than the simple reactive behavior requires. The philosophy of the approach affects the performance of planning.

4. Types of Representation Available

All Representation Spaces are acquired from the external reality by the processes of Learning. Many types of learning are mentioned in the literature (supervised, unsupervised, reinforcement, dynamic, PAC, etc.) Before classifying a need in a particular method of learning and deciding how to learn, we would like to figure out what exactly we should learn. Can the process of learning be separated into two different learning processes:

- that of representation, and
- that of the rules of action,

or are these two kinds of learning just two sides of the same core learning process?

The following knowledge should be contained in the Representation Space. If no GS is given, any pair of state representations should contain implicitly the rule of moving from one state to another. In this case, while learning we inadvertently consider any second state as a provisional GS.

We will call "proper" representation a representation similar to the mathematical function and/or field description: at any point the derivative is available together with the value of the function; the derivative can be considered an action required to produce the change in the value of the function.

We will call "goal oriented" representation a representation in which at each point a value of the action is given required for describing not the best way of achieving an adjacent point but the best way of achieving the final goal.

Both "proper" and "goal oriented" representations can be transformed into each other. However, they differ in the productivity of planning.

5. Components of Representation Space

Representation (that of the World) can be characterized by the following artifacts:

- existence of states with its boundaries determined by the resolution of (each state is presented as a tessellatum, or an elementary unit of representation, the lowest possible bounds of attention)
- characteristics of the tessellatum which is defined as an indistinguishability zone (we consider that resolution of the space shows how far the "adjacent" tessellata (states) are located from the "present state" (PS)
- lists of coordinate values at a particular tessellatum in space and time
- lists of actions to be applied at a particular tessellatum in space and time in order to achieve a selected adjacent tessellatum in space and time
- existence of strings of states intermingled with the strings of actions required to receive next consecutive tessellata of these strings of states
- boundaries (the largest possible bounds of the space) and obstacles
- costs of traversing from a state to a state and through strings of states.

In many cases, the states contain information which pertains to the part of the world which is beyond our ability to control it, and this part is called "environment." Another part of the world is to be controlled: this is the system for which the planning is to be performed. We will refer to it frequently as "self." Thus, a part of the representation is related to "self" including knowledge about actions which this "self" should undertake in order to traverse the environment.

It is seen from the list of artifacts that all knowledge is represented at a particular resolution. Thus, the same reality can be represented at many resolutions and the "multiresolutional representation" is presumed.

The system of representation is expected to be organized in a multiresolutional fashion. This will invoke the need in applying a number of special constraints and rules. The rules of inclusion (aggregation/decomposition) are especially important.

6. Planning in Redundant Systems

Non-redundant systems have a unique trajectory of motion from a state to a state. Redundant system is defined as a system in which there is more than one trajectory of motion from one state to another. It can be demonstrated for many realistic couples "system-environment" that

- they have a multiplicity of traversing trajectories from a IS to a GS
- these trajectories can have different costs.

These systems contain a multiplicity of alternatives of space traversal. Redundancy grows when the system is considered to be a stochastic one. The number of available alternatives grows even higher when we consider also a multiplicity of goal tessellata of a particular level of resolution under the condition of assigning the goal at a lower resolution level which is the fact in multiresolutional systems (such as NIST-RCS.)

In on-redundant systems there is no problem of planning. Since the trajectory of motion to be executed is a unique one, the problem is to find this trajectory and to provide tracking of it by an appropriate classical control system.

7. Learning as a Source of Representation

Learning is defined as knowledge acquisition via experience of functioning. Thus, learning is development and enhancement of the representation space under various goals. The representation can be characterized in the following ways:

- by a set of paths (to one or more goals) previously traversed

- by a set of paths (to one or more goals) previously found and traversed
- by a set of paths (to one or more goals) previously found and not traversed
- by a totality of (set of all possible) paths
- by a set of paths executed in the space in a random way.

One can see that this knowledge contains implicitly both the description of the environment and the description of the actions required to traverse a trajectory in this environment. Moreover, if some particular system is the source of knowledge, then the collected knowledge contains information about properties of the system which moved in the environment.

All this information arrives in the form of experiences which record states, actions between each couple of states, and evaluation of the outcome. The collection of information obtained in one or several of these ways forms knowledge of space, KS.

If the information base contains all tessellata of the space with all costs among the adjacent tessellata - we usually call it the "a complete representation." The tessellation can be a randomized one: a factor strongly affecting the performance.

Thus, the representation is equivalent to the multiplicity of explanations how to traverse, or how to move. In other words: all kinds of learning mentioned in p. 3 are equivalent.

Comments: a) Knowledge of the space (KS) is realized via knowing states, and/or knowing the "derivatives" (or actions) from a state to a state.

b) Apparently, each state can be characterized by some cumulative cost (value), while each traversal from a state to a state can be characterized by some incremental cost (goodness of a move or a set of moves.)

8. Standardizing the Problems of Planning

Any problem of planning is associated with

- actual existence of the present state
- actual, or potential existence of the goal state
- knowledge of the values for all or part of the states as far as some particular goal is concerned.

From this knowledge the cumulative costs of trajectories to a particular goal (or goals) can be deduced. On the other hand, the knowledge of costs for the many trajectories traversed in the past can be obtained which is equivalent to knowing cumulative costs from the initial state (PS) to the goal state (GS) (from which the values of the states can be deduced.)

In other words, any problem of planning contains two components: the first one is to determine and/or to refine the goal (bring it to the higher resolution.) The second one is to determine the motion trajectory to this refined goal. These two parts can be performed together, or separately. Frequently we are dealing with them separately. In the latter case they are formulated as follows:

a) given PS, GS and KS (all paths) find the subset of KS with a minimum cost, or with a pre-assigned cost, or with a cost in a particular interval.

b) given PS and GS from the lower resolution level and KS (all paths) find the GS with a particular value (which is satisfactory for the system).

9. Performance of Planning Algorithms

Finding solutions for these problems is done by a process that is called *planning*. In other words, planning is construction of the goal states, and/or strings of states connecting the present state with the goal states. Performance of planning algorithms is determined by the way these procedures are arranged.

The first component of the planning algorithm is translation of the goal state description from the language of low resolution to the level of high resolution. Frequently, it is associated with increasing of the total number of the state variables. In all cases it is associated with reduction of the indistinguishability zone, or the size of the tessellatum associated with a particular variable.

The second component is the simulation of all available alternatives of the motion from the initial state, IS to one or several goal states, GS and selection of the "best" trajectory. Procedurally, this simulation is performed as a search, i.e. via combinatorial construction of all possible strings (groups). To make this combinatorial search for a desirable group more efficient we reduce the space of searching by focusing attention.

Thus, all planning algorithms consist of two components: a) a module for exploration of spatial distribution of the trajectory, and b) a module for exploration of the temporal distribution. No algorithm of planning is conceivable without these two components.

The need in planning is determined by the multialternative character of the reality The process of planning can be made more efficient by using appropriate heuristics which is not considered in this paper.

10. The Relations Between Planning and Learning

Planning is learning from experience in the domain of imagination. Planning is performed by searching within a limited subspace

- for a state with a particular value (designing the goal)
- for a string (a group) of states connecting SP and GP satisfying

some conditions on the cumulative cost (planning of the course of actions)

The process of searching is associated ether with collecting the additional information about experiences, or with extracting from KS the implicit information about the state and moving from state to state, or learning. In other words, planning is inseparable from and complementary to learning.

This unified planning/learning process is always oriented toward improvement of functioning in engineering systems (improvement of accuracy in an adaptive controller) and/or toward increasing of probability of survival (emergence of the advanced viruses for the known diseases that can resist various medications, e.g. antibiotics.)

Thus, this joint process can be related to a system as well as to populations of systems and determines their evolution.

11. Other Components of Planning

Planning algorithms consist of the procedures of Job Assignment and Scheduling. Job Assignment distributes the motion among the spatial coordinates. Scheduling distributes the motion along the time axis. Together, they contribute to the search process. Search is performed by constructing feasible combinations of the states within a subspace. ("Feasible" means: satisfying a particular set of conditions.) Search is interpreted as exploring (physically, or in simulation) as many as possible alternatives of possible motion and comparing them afterwards.

Each alternative is created by using a particular law of producing the group of interest (cluster, string, etc.) Usually, grouping presumes exploratory construction of possible combinations of the elements of space (combinatorial search) and as one or many of these combinations satisfy conditions of "being an entity" - substitution of this group by a new symbol with subsequent treating it as an object (grouping.)

The larger the space of search is the higher is the complexity of search. This is why a special effort is allocated with reducing the space of search. This effort is called focusing attention and it results in determining two conditions of searching, namely, its upper and lower boundaries:

a) the upper boundaries of the space in which the search should be performed, and

b) the resolution of representation (the lower boundaries)

12. Planning Embodies the Intelligence of a System

Formation of multiple combinations of elements (during the search procedure, S) satisfying required conditions of transforming them into entities (grouping, G) within a bounded subspace (focusing attention, F) is a fundamental procedure in both learning and planning. Since these three procedures work together we will talk about them as about a triplet of computational procedures which include grouping, focusing attention and search (GFS.) Notice, that in learning it creates lower resolution levels out of higher resolution levels (bottom-up) while in planning it progresses from the lower resolution levels out of higher resolution levels (top-down.)

This triplet of computational procedures is characteristic for intelligence and probably is the elementary computational unit of intelligence. Its purpose is transformation of large volumes of information into a manageable form which ensures success of functioning. The way it functions in a joint learning-planning process explains the pervasive character of hierarchical architectures in all domains of activities.

The need in GFS is stimulated by the property of knowledge representations to contain a multiplicity of alternatives of space traversal (which is a property of representations to be redundant.) Redundancy of representations determines the need in GFS: otherwise the known systems would not be able to function efficiently (it is possible that redundancy of representations is a precondition for the possibility of Life and the need in Intelligence)

13. Planning is Inseparable From Control

Development of a plan is equivalent to computing the "feedforward control." To compute FFC, we have to have a model of a system (representation) and apply an operation if inverse (computing the required FFC control commands for the motion preassigned). Even if a system representation is in a not-invertible form, the inverse can be found by a forward searching.

Representations reduce the redundancy of reality. Elimination of redundancy allows for having problems that can be solved in a closed form (no combinatorics is possible and/or necessary). Sometimes, this ultimate reduction of redundancy is impossible and the combinatorial search is the only way of solving the problem). If the problem cannot be solved in a closed form, we introduce redundancy intentionally to enable functioning of GFS (grouping, focusing attention, and searching).

At each level of resolution, planning is done as a reaction for the slow changes in situation which invokes the need in anticipation and active interference

a) to take advantage of the growing opportunities, or

b) to take necessary measures before the negative consequences occur.

The deviations from a plan are compensated for by the compensatory mechanism also in a reactive manner. Thus, both feedforward control (planning) and feedback compensation are reactive activities as far as interaction system-environment is concerned. Both can be made active in their implementation. This explains different approaches in control theory.

Examples:

a) Classical control systems are systems with no redundancy, they can be solved in a closed form. Thus, they do not require any searching.

b) Any stochastics introduced to a control system creates redundancy and requires either for elimination of redundancy and bringing the solution to a closed form, or performing search.

c) Optimum control allows for the degree of redundancy which determines the need in searching.

Recently, an area of "supervisory control" has emerged as a partial introduction of the control theory to the domain of planning.

14. Research for Planning: Topics For Exploration and Discussion

The following research topics can be outlined:

a) development of the system of representation for planning purposes; it should provide for a multiresolutional organization of information

b) analysis of existing and potentially beneficial techniques of synthesizing the goal assignments (spatial plan distribution)

--by using combinatorial techniques (computer and human-based)

--by analytical methods (e.g. variational)

c) analysis of existing and potentially beneficial techniques of determining preferable clusters, or groups: determining the preferred schedules for strings of the way points, or milestones (temporal plan distribution)

--by using search in the state space

--by using game-theoretical methods

--by using self-organization of multiple agents

d) quantitative evaluation of the tools for narrowing attention: determining envelopes around the trajectory of motion (string of the milestone events) for the consecutive refinement (repetition of the planning procedure at the higher resolution level)

e) construction of the state spaces for the consecutive searching

f) analysis of the methodologies of state-space tessellation for applying different methodologies of consecutive refinement

g) exploring the methods of search applicable for determining the preferential strings

--by searching techniques induced by dynamic programming

--by standard techniques of exhaustive search

--by methods of heuristic guiding during the search

--by searching via evolutionary programming

--by searching in nonlinear problems

h) testing the results of planning via various simulation methodologies

i) exploring the phenomenon of nestedness of plans obtained at various resolutions (at various levels of resolution)

j) dealing with uncertainties of information

--by decision-making procedures when the values of alternatives are uncertain and do not allow for an unequivocal choice

--by development and maintenance of contingency plans

k) analyze the role of prediction in planning, develop a system of creating and using predictions

l) analyze the phenomenon of goal

m) determine methods of forming different functionals of "cost", or "goodness"

n) explore planning under condition of multiple criteria (costs)

o) test benefits and deficiencies of various schemes of decision-making in planning

p) the computer aspects of planning are virtually unexplored: do we need a language for planning?

All positions of this list affect the performance both of the system in the World and of planning algorithms as a part of the Design process.

Lower Bounds for Evaluating Schedule Performance in Flexible Job Shops

Imed Kacem, Slim Hammadi and Pierre Borne

Abstract

In this paper, we are interested in the multiobjective evaluation of the schedule performance in the flexible job shops. The Flexible Job Shop Scheduling Problem (FJSP) is known in the litteratue as one of the hardest combinatorial optimization problems and presents many objectives to be optimized. In this way, we aim to determine a set of lower bounds for certain criteria which will be able to characterize the feasible solutions of such a problem. The studied criteria are the following: the makespan, the workload of the critical machine, and the total workload of all the machines. Our study relates to the determination of a practical method in order to evaluate the representative performance of the production system.

Keywords

Performance evaluation, lower bounds, flexible job shop scheduling problems, multiobjective optimization.

I. Introduction

The flexible job shop scheduling problem is a problem of planning and organization of a set of tasks to be performed on a set of resources with variable performances [1], [2], [3], [4], [5]. In the literature, the authors generally consider two steps in its resolution [4]. The first one is to assign the various tasks to the suitable resources. The second step is the sequencing of the tasks and the computation of the starting times by taking account of the various constraints of precedence and resources. Nevertheless, Dauzère-Pérès et al have recently proposed an interesting method to solve this problem by considering the two steps at the same time [5]. Morover, in [6], Dauzère-Pérès et al have given an extension of this problem in which tasks can be performed by several resources at the same time. Many criteria can be considered in the resolution of such a problem. Mainly, two objectives could be distinguished. The first one is to balance the workloads of the machines, the second one is to organize the various tasks in minimizing the overall completion times. This variety of criteria induces additionnal difficulties related to the evaluation of the feasible solutions as well as the comparison between the resolution methods.

In this way, we aim in this paper to contribute in the reduction of such difficulties by proposing a set of lower bounds for some representative criteria of such a problem. In addition, we propose to undervalue the criteria related to the workload of the resources. The second part of this article presents the specificities of the flexible job shops and gives the mathematical formulation used to deal with such a problem. Thus in section III, we describe the various steps followed in the calculation of the lower bounds proposed. The last part will be devoted to the presentation of some results and some conclusions concerning this research work.

II. Problem Formulation

The problem is to organize the execution of N jobs on M machines. The set of machines is noted U. Each job J_j represents a number of n_j non preemptable ordered operations (precedence constraint). The execution of the i^{th} operation of job J_j (noted $O_{i,j}$) requires one resource or machine selected from a set of available machines. The assignment of the operation $O_{i,j}$ to the machine M_k entails the occupation of this machine during a processing time called $d_{i,j,k}$. Thus, to each FJSP (Flexible Job Shop Scheduling Problem), we can associate a table D of processing times such that: $D = \{d_{i,j,k} \in IN^* \mid 1 \leq j \leq N; 1 \leq i \leq n_j; 1 \leq k \leq M\}$.

In this problem, we make the following hypotheses:

- all machines are available at $t = 0$ and each job J_j can be started at $t = r_j$,

- at a given time, a machine can only execute one operation: it becomes available to other operations once the operation which is currently assigned to is completed (resource constraints),

- to each operation $O_{i,j}$, we associate an earliest starting time $r_{i,j}$ calculated by the following formula:

$r_{1,j} = r_j \ \forall\, 1 \leq j \leq N$, and $r_{i+1,j} = r_{i,j} + \gamma_{i,j} \ \ \forall\, 1 \leq i \leq n_j - 1, \forall\, 1 \leq j \leq N$.

where $\gamma_{i,j} = \min_k (d_{i,j,k}) \quad \forall\, 1 \leq i \leq n_j - 1, \forall\, 1 \leq j \leq N$.

The FJSPs present two difficulties. The first one is to assign each operation $O_{i,j}$ to a machine M_k (selected from the set U). The second one is the computation of the starting time $t_{i,j}$ and the completion time $tf_{i,j}$ of each operation $O_{i,j}$.

The considered objective is to minimize the following criteria:

- the makespan:

The authors are with "Laboratoire d'Automatique et Informatique de Lille", at "Ecole Centrale de Lille", Cité scientifique BP 48. Villeneuve d'Ascq, 59651, France. (Corresponding author: imed.kacem@ec-lille.fr).

$$Cr_1 = \max_j \{tf_{n_j,j}\} \tag{1}$$

- the workload of the most loaded machine:

$$Cr_2 = \max_k \{W_k\} \tag{2}$$

where W_k is the workload of M_k,
- the total workload of the machines:

$$Cr_3 = \sum_k W_k \tag{3}$$

Definition 1: To each repartition of the operations on the resources set, we associate an assignment. Each assignment is characterized by a set S such that: $S = \{S_{i,j,k} \in \{0,1\} \,|\, 1 \leq j \leq N, 1 \leq i \leq n_j, 1 \leq k \leq M\}$. $S_{i,j,k} = 1$ if $O_{i,j}$ is assigned to M_k else $S_{i,j,k} = 0$.

Definition 2: $\widetilde{E}$ is a numerical function defined as follows: $\widetilde{E}(x) = x$ if x is integer else $\widetilde{E}(x) = E(x) + 1$, where $E(x)$ is the integer part of x.

III. New Lower Bounds

The problem of the lower bounds has been considered in the literature for many scheduling problems, in particular, the one-machine problem [7], the parallel machines problrm [8], [9], the hybrid flow-shop problem [11] and the job-shop problem [12]. Generally, the methods suggested are based on the constraint relaxation (preemption of the tasks, disjunctive constraint on the resources...) in order to estimate the makespan of the optimal schedule. In this paper, we generalize some results proposed in the literature for the parallel machines problem [9] and we propose others new considerations based on the evaluation of the cost of assigning a certain number of tasks to certain resources.

A. Lower Bound for the Workload of the Machines

The total workload is equal to the sum of all the processing times of all the operations carried out by the set of the machines according to the chosen assignment. In addition, for each operation $O_{i,j}$, the processing time is superior or equal to $\gamma_{i,j}$, therefore:

$$Cr_3 \geq \sum_j \sum_i \gamma_{i,j}$$

B. Lower Bound for the Workload of the Critical Machine

Lemma 1: $\widetilde{E}\left(\frac{\sum_j \sum_i \gamma_{i,j}}{M}\right)$ is a lower bound of Cr_2.

Proof: Trivial, the workload of the critical machine is higher than the mean of the workloads. ■

Now, let consider the operations carried out by the set of the machines. $\widehat{N}$ is the average of the operation numbers carried out by one machine ($\widehat{N} = \frac{1}{M}.\sum_j n_j$). Then, we have at least one machine M_{k_0} that will perform at least N_{k_0} operations such that: $N_{k_0} \geq \widetilde{N} = \widetilde{E}\left(\widehat{N}\right)$. Thus, to each machine M_k, we associate the $\widetilde{N}$ shortest operations that it can perform. $D_{k,\widetilde{N}}$ is the sum of the processing times of these associated operations and $\delta_{k_0,\widetilde{N}}$ represents the minimal value of these sums when varying k:

$$\delta_{k_0,\widetilde{N}} = \min_{1<k<M} \left(D_{k,\widetilde{N}}\right)$$

The critical machine will have a workload more important than $\delta_{k_0,\widetilde{N}}$, then, we obtain the following lemma.

Lemma 2: $\delta_{k_0,\widetilde{N}}$ is a lower bound of Cr_2.

Definition 3: $a(k, N')$ is a binary variable which is true if the machine M_k carries out N' operations among the NT operations (NT is the total number of the operations and $N' \leq NT$).

Let us suppose that $a(k, N') = 1$, then the machine M_k will work during a time at least equal to $D_{k,N'}$ ($D_{k,N'}$ is the sum of the N' shortest processing times of the operations that we can perform on M_k). In this case, the remainder of the operations (the $NT - N'$ operations) must be carried out on $U^k = U - \{M_k\}$. This obligation can be regarded as a scheduling problem of $(NT - N')$ operations on the $(M-1)$ machines of U^k. Therefore, for such a problem, we can apply the result of lemma 1 to undervalue the workload of the most loaded machine of U^k. Thus, this workload is superior or equal to $\alpha_{k,N'} = \widetilde{E}\left(\frac{\Gamma_{k,N'}}{M-1}\right)$, where $\Gamma_{k,N'}$ is the sum of the $(NT - N')$ smallest values of $\gamma^k_{i,j}$ such that $\gamma^k_{i,j} = \min_{\substack{1<h<M \\ h \neq k}} (d_{i,j,h})$. Then, we obtain the following lemma.

Lemma 3: If $a(k, N') = 1$, then there exists a machine for which the workload is at least equal to $\beta_{k,N'}$ such that:

$$\beta_{k,N'} = \max\left(D_{k,N'}, \alpha_{k,N'}\right)$$

Lemma 4: $\mu_{k_0,\widetilde{N}} = \min_{1<k<M}\left(\min_{N' \geq \widetilde{N}}\left(\beta_{k,N'}\right)\right)$ is a lower bound of Cr_2 and it is superior to $\delta_{k_0,\widetilde{N}}$.

Proof: Obvious, according to lemma 3 and the definition of $\widetilde{N}$. ∎

Using lemmas 1 and 4, we deduce the following lemma.

Lemma 5:

$$Cr_2 \geq \max\left(\widetilde{E}\left(\frac{\sum_j \sum_i \gamma_{i,j}}{M}\right), \mu_{k_0,\widetilde{N}}\right)$$

C. Lower Bound for the Makespan

Lemma 6:

$$Cr_1 \geq \max_j\left(r_j + \sum_i \gamma_{i,j}\right)$$

Proof: Obvious, because of precedence constraints on the operations over the jobs. ∎

Lemma 7: Let R_M the sum of the M smallest release dates $r_{i,j}$ of the operations, then:

$$Cr_1 \geq \widetilde{E}\left(\frac{R_M + \sum_j \sum_i \gamma_{i,j}}{M}\right)$$

Proof: This demonstration is inspired of [9]. In 1987, Carlier have proposed an interesting lower bound for the problem of n operations on m identical machines in minimizing the makespan. This bound takes account of the release date of each operation and shows that the activity intervals of the machines, in the best case, have the following form: $[r_{i_k}, f^*]\ \forall\ 1 \leq k \leq m$ where f^* is the optimal value of the makespan and $[r_{i_1}, r_{i_2}, ..., r_{i_m}]$ the m smallest release dates of the operations. By using the same idea in our problem, we obtain the following relation: $Cr_1 \geq \frac{R_M + Cr_3}{M}$. Moreover, we have $Cr_3 \geq \sum_j \sum_i \gamma_{i,j}$ and the makespan is integer, therefore, the lemma is justified. ∎

Definition 4: Let $E_{N'}$ the set of the combinations constituded of N' operations among the NT operations and $C_{N'}$ an element of $E_{N'}$: $C_{N'} = \{O_{i_1,j_1}, O_{i_2,j_2}, ..., O_{i_{N'},j_{N'}}\}$. We define $T_{k,C_{N'}}$ as follows:

$$T_{k,C_{N'}} = \min_{1<q<N'}\left\{r_{i_q,j_q}\right\} + \sum_{1<q<N'} d_{i_q,j_q,k}$$

Proposition 1: $T_{k,N'}$ is a lower bound of the value of the date at which the machine M_k can carry out N' operations.

$$T_{k,N'} = \min_{C_{N'} \in E_{N'}}\left\{T_{k,C_{N'}}\right\}$$

Proof: For a given combination $C_{N'}$ of $E_{N'}$, the machine M_k can start to perform operations at least at the date $t = \min_{1<q<N'}\left\{r_{i_q,j_q}\right\}$ and must work for a duration at least equal to $\sum_{1<q<N'} d_{i_q,j_q,k}$. ∎

Lemma 8: Let V_z the subset of the operations set defined as follows: $V_z = \left\{O_{i_{z+1},j_{z+1}}, O_{i_{z+2},j_{z+2}}, ..., O_{i_{NT},j_{NT}}\right\}$ such that $r_{i_{z+1},j_{z+1}} \leq r_{i_{z+2},j_{z+2}} \leq ... \leq r_{i_{NT},j_{NT}}$, then:

$$T_{k,N\prime} = \min_{1<z<NT-N\prime+1}\left(r_{i_z,j_z} + d_{i_z,j_z,k} + \min_{C'_{z,N\prime}\in E'_{z,N\prime}}\left(\Delta^k\left(C'_{z,N\prime}\right)\right)\right)$$

where: $\Delta^k\left(C'_{z,N\prime}\right)$ is the sum of the processing times of the operations of $C'_{z,N\prime}$ on M_k; $C'_{z,N\prime}$ is an element of $E'_{z,N\prime}$ and $E'_{z,N\prime}$ is the set of the combinations of $(N\prime-1)$ operations chosen among the $(NT-z)$ operations of V_z for $z \in \{1, 2, ..., N_t - N\prime + 1\}$.

Proof: For proof, see [10]. ■

Let us suppose that $a(k, N\prime) = 1$, then the machine M_k will work until a date at least equal to $T_{k,N\prime}$. In this case, the remainder of the operations (the $NT - N\prime$ operations) must be carried out on U^k. This obligation can be regarded as a scheduling problem of $(NT - N\prime)$ operations on the $(M-1)$ machines of U^k. Therefore, for such a problem, we can apply the result of lemma 7 to undervalue the necessary time for such an execution (the minorant of such necessary time is noted $\lambda_{k,N\prime}$). Thus, if $NT - N\prime \geq M - 1$, $\lambda_{k,N\prime} = \widetilde{E}\left(\frac{R_{M-1}+\Gamma_{k,N\prime}}{M-1}\right)$ where R_{M-1} is the sum of the $(M-1)$ smallest values of $r_{i,j}$. In the contrary case and if $NT - N\prime < M - 1$, it is more interesting to undervalue the time in question by $\widetilde{E}\left(\frac{R_{NT-N\prime}+\Gamma_{k,N\prime}}{NT-N\prime}\right)$ where $R_{NT-N\prime}$ is the sum of the $NT - N\prime$ smallest values of $r_{i,j}$. Therfore, we have the following lemma.

Lemma 9: If $a(k, N\prime) = 1$, then we need to perform operations until a date at least equal to $\tau_{k,N\prime}$ such that:

$$\tau_{k,N\prime} = \max\left(T_{k,N\prime}, \lambda_{k,N\prime}\right)$$

with:

$$\lambda_{k,N\prime} = \widetilde{E}\left(\frac{R_{M-1}+\Gamma_{k,N\prime}}{M-1}\right) \text{ if } NT - N\prime \geq M-1,$$

$$\lambda_{k,N\prime} = \widetilde{E}\left(\frac{R_{NT-N\prime}+\Gamma_{k,N\prime}}{NT-N\prime}\right) \text{ if } NT - N\prime < M-1.$$

Theorem 1: $\tau_{k_0,\widetilde{N}}$ is a lower bound of Cr_1:

$$\tau_{k_0,\widetilde{N}} = \min_{1<k<M}\left\{\min_{N\prime\geq\widetilde{N}}\left\{\tau_{k,N\prime}\right\}\right\}$$

Proof: $\min_{N\prime\geq\widetilde{N}}\left\{\tau_{k,N\prime}\right\}$ represent a lower bound of the makespan if the machine M_k carries out a number of operations superior or equal to $\widetilde{N}$. In addition, there is at least one machine that verify such a condition, therefore, the theorem is justified. ■

Let V the set of all the operations classified in the ascending order according to the values of $r_{i,j}$:

$$V = \left\{O_{i_1,j_1}, O_{i_2,j_2}, ..., O_{i_{NT},j_{NT}}\right\} \text{ such that } r_{i_1,j_1} \leq r_{i_2,j_2} \leq ... \leq r_{i_{NT},j_{NT}}$$

It is clear that any lower bound of the scheduling problem of V_z is also a lower bound for the initial problem (the scheduling of V).

Lemma 10: $LB2$ is an improvement of the lower bound proposed in lemma 7:

$$LB2 = \max_{1<q<NT-M}\left\{\frac{1}{M}\cdot\left(\sum_{h=q}^{h=q+M-1} r_{i_h,j_h} + \sum_{h=q}^{h=NT}\gamma_{i_h,j_h}\right)\right\}$$

Proof: Such a result is justified by applying lemma 7 to the subsets V_z for $1 \leq z \leq NT - M - 1$. ■

Corollary 1 : Using the preceding theorems and the preceding lemmas, we obtain the folowing relation:

$$Cr_1 \geq \max\left(\max_j\left(r_j + \sum_i \gamma_{i,j}\right), LB2, \tau_{k_0,\widetilde{N}}\right)$$

*Remark*1 : In a previous work [13], we have showed that it exists an equivalence between a flexible job-shop with release dates and a flexible job-shop without release dates. Thus, we can use this equivalence to find other relations as the following relation:

$$Cr_1 \geq \max\left(\widetilde{E}\left(\frac{\sum_j r_j + \sum_j \sum_i \gamma_{i,j}}{N+M}\right), \delta_{k_0,\widetilde{N}\prime}\right)$$

with $\widetilde{N}\prime = \widetilde{E}\left(\frac{N+NT}{N+M}\right)$, $\delta_{k_0,\widetilde{N}\prime} = \min_{1<k<M}\left(D_{k,\widetilde{N}\prime}\right)$ and $D_{k,\widetilde{N}\prime}$ the function previously defined in the preceding subsection, but one can easily show that such bounds are less interesting than those defined by the preceding corollary.

D. Recapitulation

The preceding minorations will enable us to compute some limits for the values corresponding to the three criteria considered. These limits are defined by the following relations:

$$Cr_1 \geq Cr_1^*, Cr_2 \geq Cr_2^* \text{ and } Cr_3 \geq Cr_3^*$$

with:

$$Cr_1^* = \max\left(\max_j\left(r_j + \sum_i \gamma_{i,j}\right), LB2, \tau_{k_0,\widetilde{N}}\right),$$

$$Cr_2^* = \max\left(\widetilde{E}\left(\frac{\sum_j \sum_i \gamma_{i,j}}{M}\right), \mu_{k_0,\widetilde{N}}\right)$$

$$\text{and } Cr_3^* = \sum_j \sum_i \gamma_{i,j}.$$

E. Complexity

The formulas of the different lower bounds are implemented according to the corresponding algorithms. All these algorithms are polynomial. Theirs algorithmic complexities are presented in Table I.

IV. Simulation Results

To test the efficiency of the lower bounds in the evaluation of the system performance, many computational experiments have been carried out. This test consists in applying a Controlled Evolutionary Approach (CEA) [2]. The objective of such simulations is not to evaluate the efficiency of the CEA (in fact, the performance of such a method has been demonstrated in previous publications [2][14]). The objective considered is to measure the quality of the lower bounds proposed by comparing them to the various values of the criteria associated to the solutions given by the evoked method. In this section, we give a short description of the CEA, then, we present the considered examples and we finish by comparing the criteria values of the obtained solutions to the lower bounds values.

TABLE I

LOWER BOUNDS COMPLEXITIES

Lower bound	Complexity
Cr_1^*	$O(NT^2)$
Cr_2^*	$O(NT^2)$
Cr_3^*	$O(NT)$

A. Used Method

The problem considered presents two main difficulties. The first one is the assignment of each operation to the suitable machine. The second difficulty is the calculation of the starting times $t_{i,j}$ of each operation $O_{i,j}$. To solve such a problem, we apply initially an Approach by Localization [2]. This approach is a heuristic which makes it possible to assign the operations to the machines by taking account of the processing times and the workloads of the machines to which we have already assigned operations. Then, it makes it possible to solve the problem of the tasks sequencing thanks to an algorithm called "Scheduling Algorithm" [2] which calculates the starting times $t_{i,j}$ by taking into account the availabilities of the machines and the precedence constraints. The conflicts are solved by using traditional priority rules (SPT, LPT, FIFO, LIFO, FIRO [15], [16]), thus, we obtain a set of schedules according to the used priority rules. To such a set, we apply an evolutionary approach which is based on the schemata theorem introduced in the genetic algorithms field. Such an approach consists in the design of an assignment model which will be useful to construct the set of the new individuals. The objective is to integrate the good qualities contained in the schemata [2], [17], [18] in order to make the evolutionary algorithm more effecient and more rapid. In fact, the construction of the solutions is done by giving the priority to the reproduction of the individuals respecting the model generated by the assignment schemata and not starting from the whole set of the chromosomes (for further details, the reader is invited to consult [2] and [18]). The multiobjective evaluation of the solutions is carried out using a fuzzy Pareto approach [14]. Such an approach is based on the weighted aggregation of the different objective functions at each iteration of the evolutionary algorithm. The particularity of such an approach consists in the fuzzy computation of weights by giving the priority to the objective functions which the values are far from the corresponding lower bound value [14].

B. Results

Many series of examples have been tested to evaluate the quality of the lower bounds based on practical data. As an example, the reader could consult the simulation results of some instances at the web address: http://www.ec-lille.fr/~kacem/testsPareto.pdf. These instances come from the literature [4], [14] and present problems with 4 to 25 jobs, generally using 10 machines with 12 to 75 operations with total flexibility. The various results are summarized in Table II. For each instance, we present the values of the different lower bounds and the values of the criteria for the Pareto optimal solutions obtained by our fuzzy evolutionary approach.

The results presented in Table II show that the solutions obtained are generally very close to the optimal one. In the case of the Parallel Machines problem (a particular case of the flexible job-shop described by the instances I_5, I_6, I_7, I_8, I_9 and I_{10}), we obtain no distance between the lower bounds values and the different obtained solutions. In the general case, the small distance that we can have is due to the difficulty of multiobjective optimization which considers several nonhomogeneous and antagonistic criteria at the same time [19]. Recently, we have also tested some instances coming from the benchmarks of Hurink [20]. The results obtained confirm the good quality of the proposed lower bounds.

V. Conclusion

In this paper, we have proposed a set of lower bounds to make it easier the multiobjective evaluation problem of a schedule performance in the case of flexible job shops. These lower bounds make it possible to estimate precise limits for the optimal values of the corresponding criteria. In fact, different simulations show that the little distance between such limits and the values of the criteria obtained for the solutions generated by the evolutionary fuzzy approach is generally satisfactory and promising. This result is very interesting to facilitate the study of others multiobjective concepts (Uniform Design concept [19]) that we will consider as perspective in our future work.

Acknowledgement : We would like to thank Professor Jacques Carlier for giving us some interesting references related to his work on lower bounds for some scheduling problems.

References

[1] P. Brandimarte, Routing and scheduling in Flexible job shops by tabu search, Annals of Operations Research 41 (1993) 157-183.
[2] I. Kacem, S. Hammadi, P. Borne, Approach by Localization and Multi-objective Evolutionary Optimization for Flexible Job-Shop Scheduling Problems. IEEE Transactions on Systems, Man, and Cybernetics, PART C, Vol 32, N°1, pp1-13, 2002.
[3] P. Brucker, J. Neyer, Tabu search for the multi-mode job-shop problem, ORSpektrum 20, 1998, p 21-28.
[4] K. Mesghouni, Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de production, Thèse, USTL, 5 janvier, 1999, France.
[5] S. Dauzère-Pérès, J. Paulli, An integrated approach for modelling and solving the general multiprocessor job-shop scheduling problem using tabu search, Annals of Oper. Res., 70 (1997), pp. 281-306.
[6] S. Dauzère-Pérès, W. Roux, J.B. Lasserre, Multi-Resource Shop Scheduling Problem with Resource Flexibility, EJOR, n°107, pp. 289-305, 1998.
[7] J. Carlier, The one-machine sequencing problem, European Journal of Operational Research, 11 (1) (1982), pp 42-47.

TABLE II

SOME SIMULATION RESULTS

Instances $I_{h\ h<10}$	Lower-bounds Cr_1^*	Cr_2^*	Cr_3^*	Obtained results Cr_1	Cr_2	Cr_3
I_1	16	7	32	18	8	32
				18	7	33
				16	9	35
				16	10	34
I_2	15	9	60	15	11	61
				16	10	66
				16	12	60
				17	10	64
				18	10	63
I_3	23	10	91	24	11	91
				23	11	95
I_4	7	5	41	7	5	45
				8	7	41
				8	5	42
I_5	17	13	63	17	13	63
I_6	26	26	126	26	26	126
I_7	51	51	252	51	51	252
I_8	16	13	63	16	13	63
I_9	38	38	189	38	38	189
I_{10}	63	63	315	63	63	315

[8] B. Jurisch, Scheduling Jobs in Shops with Multi-purpose Machines, Ph.D thesis, Fachbereich Mathematik/Informatik, Universitat Osnabruck, 1992.

[9] J. Carlier, Scheduling jobs with release dates and tails on identical machines to minimize the makespan", European Journal of Operational Research, 29 (1987) 298-306, North-Holland.

[10] I. Kacem, S. Hammadi, P. Borne, Bornes Inférieures pour les Problèmes d'Ordonnancement des Job-shop Flexibles, CIFA'02, Juillet, 2002, Nantes, France.

[11] J.C. Billaut, J. Carlier, E. Néron, Ordonnancement d'ateliers à ressources multiples. Ordonnancement de la production, Edition Hermès, 2002, France.

[12] J. Carlier, An algorithm for solving the job shop problem. Management science, Vol. 35, 1989, p164-176.

[13] I. Kacem, S. Hammadi, P. Borne, Approche évolutionniste modulaire contrôlée pour le problème du type job-shop flexible, Proceedings of JDA'2001, Journées Doctorales d'Automatique, 25-27 septembre 2001, Toulouse, France (in french) (available at the web address : http://www.laas.fr/JDA2001/Actes/kacem.pdf).

[14] I. Kacem, S. Hammadi, P. Borne, Pareto-optimality Approach for Flexible Job-shop Scheduling Problems: Hybridization of Evolutionary Algorithms and Fuzzy Logic. Journal of Mathematics and Computers in Simulation, Elsevier, 2002.

[15] D. Boucon, Ordonnancement d'Atélier : aide au choix de règles de priorité, Thèse ENSAE, Toulouse, 1991, France.

[16] G. Bel, J-B. Cavaillé, Approche simulatoire, Chapitre 6 de : Ordonnancement de la production, Sous la direction de P. Lopez et de F. Roubellat, Hermès, 2001. France.

[17] M-C. Portmann, Study on Crossover Operators Keeping good schemata for some scheduling problems, Genetic and Evolutionary Computation Conference, Las Vegas, 8-12 juillet 2000, USA.

[18] L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New-York, 1990, USA.

[19] Y. W. Leung, Y. Wang, Multiobjective Programming Using Uniform Design and Genetic Algorithm, IEEE/SMC Transactions, Part C, Vol 30, August 2000, pp 293-303.

[20] I. Hurink. B. Jurisch. M. Thole, Tabu search for the job-shop scheduling problem with multi-purpose machines, OR-Spektrum 15, pp 205-215.

PERFORMANCE CHARACTERISTICS OF PLANNING ACTORS

WOUT VAN WEZEL AND RENÉ JORNA

Faculty of Management and Organization
University of Groningen, P.O.Box 800
9700 AV Groningen, The Netherlands
E-mail: w.m.c.van.wezel@bdk.rug.nl; r.j.j.m.jorna@bdk.rug.nl

ABSTRACT. Planning is a field of interest in many scientific disciplines. These scientific areas cover a multitude of planning approaches that at first sight do not have much in common: psycho-physiological analyses, organizational science, linguistics, cognitive science, operations research, and spatial science, to name just a view. The differences in ontologies and methods used make it difficult to make statements that transcend the mono-disciplinary perspectives. Still, no matter the research field, planning always concerns anticipating on the future and determining courses of action. As a consequence, there must also be similarities between the various approaches that deal with planning. This paper proposes a number of characteristics that can be used to analyze the differences and similarities in performance of different kinds of planning actors.

1. INTRODUCTION

Where will we go and how do we get there? This question is an inherent part of intelligent systems. The ability to anticipate and plan is usually seen as a required and perhaps even essential feature of such systems. It is the fundament of goal directed behavior; systems that pursue goals need to take the future into account.

Planning is not a nicely ordered and well-defined subject. Various disciplines with various scientific backgrounds deal with planning, such as (cognitive) psychology, mathematics, economics, operations research, artificial intelligence, and management and organization. In our opinion, comparing, combining, or even integrating the research efforts of each of the individual planning approaches can be a fruitful next step in planning research in general. Due to the sheer differences between these scientific areas, however, it seems difficult to make generic statements about the relation between the planning of an actor and its performance. But, the fact that planning always concerns anticipating on the future and determining courses of action might provide an opening. This notion is used in this paper to introduce a frame of reference for planning. In section 2, we provide a generic and abstract definition of planning, resulting in a discussion of four different scientific planning areas. Section 3 provides a number of generic characteristics with which the planning research areas are compared. In section 4, we draw the conclusions.

2. PLANNING ACTORS

As stated in the previous section, many research areas somehow deal with planning. In this section, we will describe four of such areas. This description will be based around our conception of planning, which can be outlined by four topics.

First, it is important to acknowledge that some entity must make the plan. Note that all kinds of entities can make plans, for example, humans, robots, computer programs, animals, organizations, etc.

Second, someone or something must execute the plan, i.e., the intended future must somehow be attained. Again, this can be done by all kinds of entities, and the planning entities need not necessarily be involved in plan execution themselves.

Third, the planning entity needs some kind of model of the future, since the future is essentially non-existent. This model should include states, possible actions of the executing entities and the effect of actions on the state they reside in, constraints, and goals. Planning and anticipation presume that such a predictive model is available, otherwise the chance that a plan can be executed as intended becomes a shot in the dark.

The *fourth* element of planning is the plan itself. The plan signifies the belief that the planning entity has in the model of the future: the actions in the plan (which are performed by the executing actor) will lead to the desired or intended future state.

The first and second topics lead to four kinds of planning actors that have their own embedding in literature:

1. Humans that plan and execute: (cognitive) psychology
2. Humans that plan but do not execute: organizational science
3. Artificial actors that plan and execute: artificial intelligence
4. Artificial actors that plan but do not execute: operations research

In the remainder of this section, we will discuss these four kinds of planning actors. The third and fourth topic can be used to analyze the performance of planning actors. This will be discussed in the next section.

Humans that plan their own activities: To have a closer look at the execution of the task, we start with cognition, where the study of planning contains the study of human (intelligent) activities (tasks). In this perspective the old definition of Miller, Pribram & Galanter is used, saying that a plan is a hierarchical process within an organism that controls series of operations [1]. Jean-Michel Hoc elaborates this definition and says that planning always involves anticipation and schematization [2]. What he means is that planning is a two-line parallel process, in which a future state is taken into account (anticipation) and in which a (stored) mental scheme can be applied if a concrete planning problem arises. Therefore, Hoc talks about bottom-up and top-down processes that are always involved in making a plan.

Because of the terminology he uses, Hoc is implicitly taking a position in the complex cognitive debate about planning. This debate involves two closely related topics. The first deals with the question whether planning is a form of problem solving [3] or whether planning and problem solving only overlap [4]. The second is about the question whether human planners work hierarchically [3] or whether they plan opportunistically or even chaotically [5]. We discuss the issues successively in greater detail.

Planning and problem solving: Newell et al. [6] describe the planning method as a part of a general problem solving technique. It consists of a reformulation of the problem in more abstract and restricted terms, its solution in a simplified problem space with another level of resolution [7], its retranslation into the original problem situation and subsequently its solution. In later papers Newell & Simon rename the planning method as problem abstraction, necessary if the problem is not solvable within its original state space [3]. Because planning as well as problem solving means searching for routes (i.e., sequences of actions that lead to a solution or a goal state), the explicit distinction between planning and problem solving disappears in the later work of Newell & Simon. Planning is just one very interesting example of the general problem solving approach. Das et al. [4] argue against this "planning is a subset of problem solving" approach in saying that a difference exists in problems to prove and problems to find. According to Das et al. [4, p. 40] "planning is a more pervasive, general regulating process than problem solving, with problem solving being a part of a planning process." Planning includes anticipation and overview and refers to future actions, whereas these components seem to be absent in problem solving. According to us this may almost be a game with words, because one could state that searching and trying to reach a goal and constructing a problem space with states and operators, imply future actions and anticipation. We will not settle the discussion, here. Das et al., however, may have a point in one aspect of this debate. An enigmatic element in the problem solving approach of Newell & Simon has always been the starting point of the problem solving process. How does a problem solver construct a problem space? Where does the choice for a particular problem space come from? Why does a problem solver constructs this special problem space and not another? In terms of Newell & Simon the question is how a task environment gets its representation in a state space description. It is easy to say that one has a new problem here, which requires a second order state space description. Although this might be true in the strict sense of the word, it does not solve the issue. Perhaps something like what Das et al. called "overview" or "having a higher perspective" is necessary [4]. Therefore, it might be insightful to distinguish planning as second order problem solving from "ordinary" problem solving. If, in line with Newell & Simon, one considers the planning task in organizations and institutions to be a problem solving process, the question appears how planners construct an initial representation. Do they start with an overview or are they just trying? In the first situation there is an explicit state space to start with. In the second situation the state space is reformulated again and again.

Hierarchical and opportunistic planning: The discussion about the relation between planning and problem solving is closely connected to the way the planning (or problem solving) procedure is carried out in practice: hierarchical, opportunistic, or even chaotic. In the first place because the suggestion may be present that solving a problem with or without an overview is done straightforward. One just has to follow a couple of rules from top to bottom and one ends up with a solution. In the second place the issue of the overlap between planning and problem solving very much depends on the format of representations in the information processing system of the human planner. Do planners use production rules? How are these rules controlled? Or do planners use schemata and frames? Both issues come together in the discussion started by Hayes-Roth & Hayes-Roth [5] about hierarchical and opportunistic planning.

Hierarchical planning means that there is a nested number of goal and sub-goal structures or a hierarchy of representations of a plan. The highest level in the hierarchy may be a simplification or an abstraction, whereas the lowest level is a concrete sequence of actions to solve (a part of) the planning problem. One solves a planning problem by starting at the highest level and then one continues by realizing sub-goals until one reaches the final solution. Hayes-Roth & Hayes-Roth relate this to a distinction in the overview and the action aspect of plans that they successively call plan-formation and plan-execution [5].

Unjustly, but quite understandably, the hierarchical approach is attributed to Newell & Simon. They started to talk about problem solving in terms of problem spaces, goal hierarchies, and universal sub-goaling. We consider this attribution to be at least partly wrong - one only has to recall Simon's bounded rationality concept - but we are not going to discuss the issue here [8].

In a contradistinction to the hierarchical view on planning, Hayes-Roth & Hayes-Roth propose a so called opportunistic approach to planning. This non-hierarchical planning assumes that a plan is created with the help of some kind of mental blackboard where pieces of information, relevant cues, and possible sub-goals are stored. They claim and show that planning happens asynchronously and is determined by the momentary aspects of the problem. No fixed order of operations exists; plan creation and the steps to be taken grow out of the problem stage at hand. When planners solve a planning problem, they may start with the top-goal, but very soon they loose track of the goal structure and then they continue to fulfill the goals that are reachable within reasonable time. The hierarchy very soon vanishes and what remains is some sort of heterarchy. Therefore, this kind of planning behavior is called opportunistic.

Although the contrast with the hierarchical approach may be large, a strong similarity is also present. In the hierarchical as well as in the opportunistic approach the fundamental assumption is that planning is problem solving, that can best be described in terms of problem spaces, production rules, and goals. That is to say that the basic descriptive structure is the same for both, but that real behavior within the problem space is executed differently.

With regard to the problem space description, hierarchical as well as opportunistic planning differ from the perspective defended by Riesbeck & Schank [9]. The representation of planning problems is described in terms of scripts and frames consisting of objects, slots, and relations. The information in the cognitive system, necessary to make a plan, is semi-hierarchically structured. This means that some kind of representational skeleton or framework is retrieved from memory. Stored plans contain guidelines for resolution of sorts of problems. In this process two stages exist. First a skeleton plan is found, and second the abstract steps in a plan are filled with concrete operations. Although general cognitive processing is involved in making a plan the emphasis in this approach is on the memory system. Plans at different levels of abstraction and in different formats are stored in and retrieved from memory. There are strong similarities with the approach to planning that Hoc proposed [2].

It is very difficult to reveal the different mental representations planners use in solving planning problems. Asking them whether they use production rules or scripts is not reliable and might also give them a cue.

From the above discussions, we derive the following conclusions. Together, the paradigms that were discussed provide various interpretations of a cognitive approach to human planning. In this approach, planning is about how to find the actions that solve a problem or, more general, reach a goal. The process of planning is not neatly hierarchical but switches in level of abstraction and in the time frame under consideration. The process itself is about formulating goals, finding similar solved goals, finding existing plans, adapting plans, and storing plans in such a way that they can easily be found for future reference.

There is another kind of planning that is performed by humans. Instead of planning ones own activities, humans can be involved in coordinating the activities of others. Typically, this takes place in organizations, and we make the shift from planning and executing your own activities to planning of activities that are executed by others.

Humans that plan organizational processes: Planning is a phenomenon that occurs at multiple places in an organization. In its most abstract sense, all activities that involve the determination of the future of the organization are dealing with planning. This includes strategic considerations that determine "where the organization must stand" in 10 years, less abstract issues such as growth targets or product innovations, but also very concrete decisions such as who will work at what time next week, or the exact production times and machine allocations of the production for the following week [10,11]. It is the type of planning about concrete entities that we primarily focus upon. That kind of planning is about coordination of activities of organizational members and the allocation of resources [12]. The types of activities and resources vary widely over organizations. A rough categorization that is based on the things that are planned is the distinction between production planning (machines, orders, machine operators), staff planning (shifts, personnel), and transportation planning (vehicles, routes, chauffeurs, shipments) [13].

Although the variety in organizational planning problems seems large, there are also many characteristics that are shared by organizations. A first generic characteristic for planning problems in organizations is that it basically concerns the coordination of supply and demand, whereby (a) the supply consists of scarce capacity and (b) the way in which this capacity is put to use can make a difference with respect to the goals in the organization [13,14,15]. Examples are producing at low costs at a production facility, having enough phone operators at a call center, or taking care that all employees work the same amount of night shifts. A second shared characteristic is that the planning process is distributed over multiple human planners. This means that plans get made in parallel, and that coordination between the plans is needed. A third shared characteristic for all organizational planning activities is that they are organized hierarchically. Planning problems in organizations are too complex to be solved by one person, so some kind of division in sub-problems is necessary. Therefore, there are approximate plans for the long term and detailed plans for the short term. This induces the need to coordinate; plans at higher hierarchical levels define the decision space for lower hierarchical levels. An example of such a hierarchy is strategic planning versus rough capacity planning versus production scheduling.

Not much literature or theory exists about the relations between the planning domain, the planning task, the organization of the planning, and the performance of plan execution. Most analyses are limited to task models, for example, McKay et al. [16], Mietus [17], Dorn [18], and Sundin [19]. Lack of a theory to explain the relation between planning complexity, planning organization, task performance, and planning support makes it is difficult to pinpoint the cause of the planners' discontent, to attribute the causes of poor organizational performance to the planning, or to analyze and design planning practices. For example, the cause of poor factory performance can be the mere impossibility of matching the requirements (e.g., there is not enough capacity available to meet the demands), the clumsiness of the organization of the planning, the inadequacy of the human planner to solve complex problems, the absence of specialized planning support in practice, or a combination of these factors.

In order to make generic statements about the planning task, it is important to know what the task performance depends upon (notice that by performance we mean execution without a qualitative connotation). According to Hayes-Roth & Hayes-Roth, the determinants of the planning task are problem characteristics, individual differences, and expertise [5]. That the task performance depends on individual differences and expertise is no surprise. This applies to all tasks. But the fact that the task performance also depends on problem characteristics leads to the statement that it is possible to describe a planning problem, at least partly, independent from the planner.

Clearly, approaches of planning your own activities deal with other questions than approaches for organizational planning. In section 3, we will analyze in what way planning of organizations differs from planning of your own actions. First, however, we will look at planning by artificial agents.

Artificial agents that plan their own activities: Artificial agents such as (simulated) robots that plan their own behavior need to be able to deal with uncertainty and incomplete information in their task environment. For such agents, planning is a means to reach the goal, just as it is with human problem solving. Due to the close resemblance of human and artificial agents, planning of artificial agents is very much related to the problem solving approaches as described earlier. Techniques from Artificial Intelligence are used to let such agents function more or less independently in their environment, and react on unforeseen events [20,21,22]. Much of the planning research in Artificial Intelligence stems from the wish to let autonomous actors or agents (such as robots) perform tasks without prescribing how the task should be carried out [23]. Most Artificial Intelligence methods, whether they are called algorithms, procedures, or heuristics, are based on state space descriptions. An agent or actor finds himself in a state, in which it can perform a limited number of actions. An action changes the state, after which it can again perform a number of actions [24]. The agent keeps on choosing and performing actions until the state it gets in somehow satisfies its goal. Planning is one way in which the agent can reach its goal. Other ways are, for example, trial and error or full search. To make a plan, an agent somehow anticipates the future by simulating the actions he will make. This requires the existence of (internal) representations. The original link to physical entities has been relinquished somewhat so planning agents are now often only computer programs that find a plan merely for the sake of research, and therefore not necessarily execute it. In this paradigm, planning is searching for a sequence of actions that will bring the agent from its current state in the goal state. Models of human problem solving, which were discussed in the previous subsection, have provided researchers in Artificial Intelligence with starting points for the planning functions of their artificial agents. Many examples are based on the initial General Problem Solver (GPS) of Newell & Simon [3], which constructs a proposed solution in general terms before working out the details, the opportunistic planning paradigm, and script-based planning. Here it becomes clear that models of human problem solving are closely related to the anticipation and planning of artificial agents.

Machines (computers) that plan organizational processes: A lot of planning research deals with automatically finding (or generating) plans for future organizational processes. Usually, this is about making a quantitative model that can search efficiently for good solutions. At first glance, the same kind of reasoning is used as in cognitive sciences: a problem space is set up and the aim is to find a state that satisfies all constraints and scores well on goal functions. The states are (just like in the cognitive problem solving approaches) transformed by operators. The difference is that states and operators comprise something else than the ones in cognitive science, namely values on variables and mathematical operations [25].

Models exist for all kinds of processes such as routing of trucks, staff scheduling, job shop scheduling [26,27], and flow shop scheduling. Some of the scientific fields that deal with this kind of research are Operations Research (e.g., linear programming, nonlinear programming, all kinds of heuristics), and Artificial Intelligence (constraint satisfaction programming, genetic algorithms). Although the approaches of course differ, they also possess common characteristics. They are based on an analysis of the entities that are scheduled. For example, to make an algorithm for a planning problem in a flow shop one must know the capacities of machines, setup- and cleaning times, the number and sizes of orders, the processing characteristics, etc. All these characteristics can be used to determine the best way to navigate through the problem space of possible solutions. An example of how such knowledge can be used in an algorithm is to start to plan on the bottleneck first, because it is often the sensible thing to do in order to avoid problems in a later stage of the planning process. Most techniques are somehow limited in the kinds of characteristics that they can handle. For example, a linear programming model cannot deal with nonlinear constraints, and temporal reasoning is tacky to implement in many mathematical techniques. Therefore, the domain analysis must be translated in the quantitative model, and the solution must be translated back to the application domain [28].

Computer programs that create schedules are rarely used on their own. The fact that information is lost during abstraction and translation of the domain into the model is widely recognized. For that reason, mathematical solution techniques are usually used in the context of decision support systems, where a planner can manipulate and change a plan manually so he is not bound to the solution that is presented by an algorithm.

As with the distinction between humans that plan for themselves and humans that plan for organizational processes, the approaches that deal with computer programs that plan for their own actions differ from approaches that deal with computer programs that plan for organizational processes. The differences have to do with the characteristics of the actors and will be analyzed in the next section.

3. Analysis of the Planning Actors

In section 2, we discussed four topics that are relevant for planning in general: (1) an entity must make the plan, (2) an entity must execute it, (3) the entity that makes the plan must have a model of the future, and (4) the plan exemplifies the ability of the planning entity to use the model of the future to lead the executing entity to the goal state. We now have nearly all the ingredients available to make a reasonable comparison between the different perspectives on planning. The goal of this comparison is to gain insight in the limitations that the approaches for the respective perspectives have and to see where those limitations come from. In the end, this should lead to a better understanding of the "planning" phenomenon, and perhaps the respective approaches can learn from each other. In section 2, we used topics 1 and 2 to describe four distinct planning areas. Topic 3 and 4 are used to assemble a number of characteristics with which the performance of planning actors can be analyzed and compared. The aspects that will be discussed in detail and that are used for comparisons are: a) the way in which the approaches deal with complexity; b) closed versus open world assumptions; c) the information processing mechanism and its architectural components such as memory and attention; d) the representations; e) communication, meaning, and interpretation; f) the characteristics of coordination; and g) aspects of execution of the plan.

Complexity reduction: Planning problems are assignment problems for which a limited set of structurally similar solutions exist. Theoretically, all solutions of a planning problem can be calculated in order to choose the best solution. Unfortunately, even the most seemingly simple planning problems are transcomputational [29], which means that enumeration of solutions is not practically possible. To overcome this, planning actors must choose a way to somehow look at a limited number of viable solutions. Indeed, many of the differences between the kinds of planning actors (see section 2) can be explained by the way in which they reduce the complexity of their planning problems [7]. Some ways in which a planning actor can deal with the complexity are:

1. *Opportunistic planning*. The planning actor takes decisions without any structure; only momentary aspects matter [5].
2. *Plan partitioning*. Plans are often multi-dimensional. The search space can be limited by first making plans for the individual dimensions, and then putting the plans together [30]. An example is a production plan that coordinates machine operators, machines, and production orders: separate plans can be made to assign machine operators to machines, and orders to the machines.
3. *Multiresolutional planning*. A plan can be made at multiple levels of resolution. The plan at a low level of granulation will have a lower complexity. This plan will constrain the search space of the plan at a higher level of granulation [7], so the total number of to be assessed alternatives is lessened.
4. *Learning*. Different plans can contain similar structures that can be reused in similar circumstances. (The abstract connotation is intended since learning can be based on a wide variety of aspects of the planning process.)

In both plan partitioning and multiresolutional planning, multiple plans are created. This decomposition must be a closed-loop process [7, p 265], i.e., the plans must together provide a complete plan. This is of special interest for multi-agent planning systems (such as an organization), where the plans can be made by different people and the coordination issue arises. Furthermore, each of these multiple plans is a (possibly complex) plan in itself, and can therefore be subject to each of these four strategies recursively.

"Closed world" vs. "open world": Looking from a generic perspective, the planning task itself can be called a synthetic or configuration task. From a task perspective realizing a suitable plan or solving a planning problem requires three nearly decomposable phases. In state space descriptions the first phase is the design of a (complex) initial state, of goal state(s) and of admissible operations to change states. The second phase is given the admissible operations to search for an (optimal) solution. In many cases search does not give an optimal solution. The most one may get is a satisfying solution and even that is often not possible. Then, the third phase starts in which one goes back to the initial state and the admissible operations and changes these in such a way that a solution is found. Formulated in other words, the phases of (1) initial state, (2) search, no solution, and (3a) start again with a new initial state follow the so-called "closed world" assumption. This is the necessary sequence if algorithms are applied. However, there is another way of dealing with the third phase which is more usual, especially if humans have to make a plan. If the second phase does not give an optimal or satisfactory outcome given the constraints and goal functions, the planner already is so much involved in the planning process, that because he has a glimpse of the solution given the constraints, he takes his "idea" of a solution for compelling. He therefore changes the initial state and the admissible operations, that is the constraints, in such a way that they fit the preconceived solution. This order of phases can be named the "open world" approach. It consists of (1) initial state, (2) search including

not finding a real or established fixed solution, and (3b) adjustment of initial state according to the "fixed" solution reality. This sequence of activities is what human planners whether in the industry or doing errands frequently and with great success do, but formalizing such knowledge for use in a computer program or robot seems to be very difficult.

Information processing mechanism and architectural components: An information processing mechanism embodies the way information is selected, combined, created, and deleted. The mechanism itself needs a physical or physiological carrier. Various possibilities are already present, such as the brain as our neurological apparatus, the layered connection system of a chip in a computer, a human individual in an organization, or a group of interconnected individuals in an organization. The most relevant distinction is the one in internal and external mechanism. With internal we mean that there is no direct access to the system from outside. Internally controlled, but not directly visible processes take place in the system. The cognitive system and the chip are internal, but they differ in the sense that the latter is designed which means that its operations are verifiable. External are information processing mechanisms such as groups of individuals or organizations. In other words, the kind of predictive model that is needed to anticipate ones own actions differs from the kind of model that is needed to anticipate actions of others. With respect to planning, this distinction is of course relevant if one realizes that if the plan needs to be communicated, a translation is necessary between the physical carrier and the receiver, which must be reckoned with during planning. This is the case when a planner makes a plan that is executed by others.

An architecture is a set of components of which the arrangement is governed by principles of form or function [21]. A cognitive architecture consists of memory components, of attention processors, of sensory and motor components, and of various kinds of central processors. The division is by function and the components are all implemented in neurological structures in the brain. Two other material structures for architectural layout are the chip and the constellation of a group of individuals. The same kind of components can be discerned for the com puter, consisting of memory, sensory and motor components, and central processors. For a group of individuals the architecture is different because the constituting elements are similar as for the individuals, but the roles and tasks are different. Again, the discussion about the character of the architecture boils down to a discussion about internally or externally defined. Internal is the cognitive architecture, whereas chips and groups of people can be dealt with externally.

(Internal) representations: In cognitive science the conceptual framework to deal with representations can be found in the approaches of classical symbol systems, connectionism, and situated action [31,32,33,34,8]. The basic idea is that humans as information processing systems have and use knowledge consisting of representations and that thinking, reasoning, and problem solving consist of manipulations of these representations at a functional level of description. A system that internally symbolizes the environment is said to have representations at its disposal. Representations consist of sets of symbol structures on which operations are defined. Examples of representations are words, pictures, semantic nets, propositions, and temporal strings. A representational system learns by means of chunking mechanisms and symbol transformations [32]. An entity that makes a plan for itself can of course misinterpret its position in the environment, for example because it cannot represent its environment or because it cannot manipulate its representation of the environment adequately. Furthermore, an entity that makes a plan for others can additionally have this problem with respect to the entities that must execute the plan. Representations are also immediately relevant for anticipation. A description of a future state in whatever symbol or sign system is the core of any discussion on anticipation. Rosen, for example, defines an anticipatory system as "a system containing a predictive model of itself and/or its environment, which allows it to change state at an instant in accord with the model's predictions pertaining to a later instant" [35]. Someone who makes a plan for an organization does not need a model of itself, but a model of others and their environment. This can complicate communication and interpretation of the to be planned system.

Communication, meaning, and interpretation: Communication means the exchange of information between different components. Depending on whether we are talking about internal or external information processing entities, communication means restrictions on the kinds of symbols or signs that are used for the exchange. If we relate this to the before mentioned discussion about representations, the various kinds of signs have different consequences. Clearly, sign notations are more powerful, but also more restricted than sign systems, which in turn are more powerful than just sign sets [36,8,37]. Unambiguous communication requires sign notations, but we know that all communication between humans is not in terms of notations. If computers require sign notations and humans work with sign systems, then if the two have to communicate, the one has to adjust to the other. Until recently, most adjustments consist of humans using notations. Now, interfaces are designed that allow computers to work with less powerful - in terms of semantic requirements -, but more flexible sign systems. This means that computers can deal with ambiguity. For mental activities no explicitness (channels, codes etc.) is necessary; for planning as an external task it is essential.

Coordination: Coordination concerns attuning or aligning various entities that are not self-evident unities. Information processing in a cognitive system is a kind of coordination mechanism (with no direct access). It is internal or mental. The coordinating processor is cognition itself. No explicit code is necessary. If the code is made explicit and obeys the requirements of a notation, then we can design an artificial intelligent agent that in its ultimate simplicity could be a chip. In case of a set of entities that not by themselves are a coherent unity, various coordination mechanism can be found, such as a hierarchy, a meta-plan, mutual adjustment, a market structure, and many others [38,39]. The important difference with the single agent is that these coordination mechanisms are external and of course with direct access.

Planning, execution and control: Making a plan, executing it, and monitoring its outcomes in reality are valued differently in planning your own actions and in planning actions of others (i.e., organizational processes). The planning in organizations usually is decoupled from the execution of the plan. There are two main reasons why the planner is someone else than the one who executes the plan. First, planning is a difficult job that requires expertise and experience. This is the organizational concept of task division. Second, a planner must be able to weigh the interests of many parties. Therefore, he must have knowledge about things that go beyond the limits of the individual tasks that are planned. The consequence of this decoupling is almost always inflexibility with respect to adaptation. For errand tasks the possible division in terms of sub-tasks may be interesting, but can in reality be intertwined with flexible adaptation after unforeseen events. If the controlling entity is itself a unity, discussions about transfer, communication, sign systems to do the communication, and representations are almost trivial. This does not make the planning task itself simpler; it only prevents the occurrence of ambiguity, interpretation, and meaning variance.

We now have discussed a number of planning approaches in section 2, and a number of generic planning characteristics in this section. Table 1 summarizes the findings. Evidently, measures to relate the performance of an actor to its planning activities are context dependent, because the differences in contexts can make performance measures incomparable. For example, the uncertainty in the task environment of an actor makes it hard to establish a clear cause-effect relation between plan and execution. Still, cross functional analyses add to our understanding of the mono-disciplines, and can help to get a better understanding of the relation between planning and performance.

4. CONCLUSION

Planning is a much debated, highly controversial, and multi-faceted issue. We stated that various kinds of actors can be discerned: natural, artificial, and collective actors. We also discussed that there is no easy exchange between the various planning approaches. Management and organization, cognitive science, mathematics, artificial intelligence, and economics, although all are discussing important issues in planning, do not start with the same problem formulation. We approached the issue of planning by looking especially at the entity that makes a plan. By looking at the kind of actors, their characteristics, and the level of description of the entities and components involved, we stated that discussions about the relation between planning and performance do not have to end in controversies and avowed misunderstandings. We have sketched the

	Natural actor		Artificial actor	
	Self planning	Organization planning	Self planning	Organization planning
Complexity reduction	Plan partitioning; Opportunistic planning; Learning	Plan partitioning; Multiresolutional planning	Plan partitioning; Learning	Plan partitioning
Close vs. open world	Fixing the reality to the solution that is found; reformulate the starting-point		Searching for a solution that fits the (modeled) reality	
Information processing mechanism	Information processing needs not to reckon with the outside world	Translation of internal internally coded information is necessary	Information processing needs not to reckon with the outside world	Translation of internally coded information is necessary; designed
Architectural components	Neurological: memory structures, attention processors		Electronic: memory structures, attention processors	Program components: procedures, variables
Representations	Self-representation	Representation of others	Self-representation	Representation of others
Communication, meaning, and interpretation		Mostly communication with sign systems or sign sets		Communication with sign notations
Coordination	Only with respect to anticipated actions	Coordination of actions of others	Only with respect to anticipated actions	Coordination of actions of others
Planning, execution, and control	Intertwined	Separated	Intertwined	Separated

TABLE 1: Characteristics of kinds of actors related to what they are planning and for whom

components and ingredients of planning actors and we showed that comparisons can be made and that positions can be clarified.

Are there good reasons to discuss planning issues in greater detail? We think there are two good reasons. The first is that any planning (or weaker: any anticipatory) system ultimately acts in an open world. There is nothing wrong with the closed world assumption, but in the end it is part of an open world. Switching between open and closed worlds is something human information processing can easily do, but it is difficult to get it realized for artificial (software) and collective systems (organizations). The second reason is that whether we like it or not, more and more of our fellow "intelligent" companions are software actors (agents) and we are interacting with them. Artificial and collective actors are also planning, but something seems to be different. This incompatibility cannot be solved by imposing one approach on all kinds of actors. It can only be realized if we know what precisely natural actors do when they make plans.

REFERENCES

[1] Miller, G.A., Galanter, E. & Pribram, K.J., *Plans and the structure of behavior*. Holt, Rinehart and Winston, 1960.

[2] Hoc, J-M., *Cognitive psychology of planning*. Academic Press: San Diego, 1989.

[3] Newell, A. & Simon, H.A., *Human Problem Solving*. Englewood Cliffs, New Jersey: Prentice-Hall, 1972.

[4] Das, J.P., Karr, B.C. & Parrila, R.K.,*Cognitive Planning*. New Delhi: Sage, 1996.

[5] Hayes-Roth, B. & Hayes-Roth, F., A cognitive model of planning. *Cognitive Science, 3*, pp. 275-310 (1979).

[6] Newell, A., Shaw, J.C. & Simon, H.A., Elements of a theory of human problem solving, *Psychological Review, 65*, 151-166 (1958).

[7] Meystel, A.M. & J.S. Albus. *Intelligent Systems: Architecture, Design, and Control.* New York: John Wiley & Sons, 2002.

[8] Jorna, R. J., *Knowledge representation and symbols in the mind.* Tübingen: Stauffenburg Verlag, 1990.

[9] Riesbeck, C.K. & Schank, R.C. *Inside case-based reasoning*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1989.

[10] Smith, S.F. & O. Lassila, Toward the Development of Flexible Mixed-Initiative Scheduling Tools. In: *Proceedings ARPA-Rome Laboratory planning initiative workshop*, Tucson, AZ, 1994.

[11] Wilkins, D.E., Can AI planners solve practical problems? *Computational Intelligence,6 (4)*, pp. 232-246 (1990).

[12] Myers, K.L., Advisable Planning Systems. In: A. Tate (ed.), *Advanced Planning Technology*, California: AAAI Press., 1996

[13] Jorna, R.J., Gazendam, H., Heesen H.C. & Wezel W. van., *Plannen en Roosteren: Taakgericht analyseren, ontwerpen en ondersteunen*. Lansa: Leidschendam, 1996. (Planning and Scheduling: task oriented analysis, design and support).

[14] Veloso, M.M., Towards Mixed-Initiative Rationale-Supported Planning. In: Tate, A. (Ed.). *Advanced Planning Technology*. Menlo Park, CA: AAAI Press, 1996.

[15] Verbraeck, A., *Developing an Adaptive Scheduling Support Environment*. Delft: Ph.D. Thesis, University of Delft, 1991.

[16] McKay, K.N., F.R. Safayeni & J.A. Buzacott, 'Common sense' realities of planning and scheduling in printed circuit board production. *International Journal of Production Research. Vol. 33, Nr. 6*, pp. 1587-1603 (1995).

[17] Mietus, D.M. *Understanding planning for effective decision support*. Groningen: Ph.D. thesis, University of Groningen, 1994.

[18] Dorn, J., Task-oriented Design of Scheduling Applications. In: J. Dorn & K.A. Froeschl (ed.). *Scheduling of production processes*. Chichester, England: Ellis Horwood. (1993).

[19] Sundin, U., Assignment and Scheduling. In: Breuker, J. & W. van der Velde (Eds.) *CommonKADS library for expertise modeling: reusable problem solving components*. Amsterdam: IOS Press, 1994.

[20] Sacerdoti, E.D., The Nonlinear Nature of Plans. *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, pp. 206-214 (1975).

[21] Curry, K.W. & A. Tate., O-Plan: The Open Planning Architecture. *Artificial Intelligence,5 1, 1* (1991).

[22] Beetz, M., Concurrent Reactive Plans. *Lecture Notes in Artificial Intelligence*. Berlin: Springer-Verlag, 2000.

[23] Fikes, R.E., & N.J. Nillson., STRIPS: a New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence,* 2, pp 189-208 (1971).

[24] Meystel, A.M., Theoretical foundations of planning and navigation for autonomous robots. *International Journal of Intelligent Systems,2* , pp. 73-128 (1987)

[25] Sanderson, P.M., The Human Planning and Scheduling Role in Advanced Manufacturing Systems: An Emerging Human Factors Domain. *Human Factors, 31 (6),* pp. 635-666 (1989).

[26] Fox, M.S., *Constraint-Directed search: A Case Study of Job-Shop Scheduling*. Ph.D. Thesis, Carnegie Mellon University, 1983.

[27] McKay, K.N., F.R. Safayeni & J.A. Buzacott,. Job-Shop Scheduling Theory: What Is Relevant? *Interfaces, 18, 4*, pp. 84-90 (1988).

[28] Prietula, M.J., W. Hsu & P.S. Ow, MacMerl: Mixed-Initiative Scheduling with Coincident Problem Spaces. In: Zweben, M. & Fox, M.S. *Intelligent Scheduling*. San Francisco: Morgan Kaufman, 1994.

[29] Klir, G.J. *Facets of systems science*. New York: Plenum Press. 1991.

[30] Wezel, W. van, *Tasks, hierarchies, and flexibility: Planning in Food processing industries.* Groningen, SOM-series, 2001.

[31] Posner, M.I. (Ed.), *Foundations of cognitive science*. Boston, Mass.: The MIT-Press, 1989).

[32] Newell, A., *Unified Theories of Cognition*. Cambridge: Harvard University Press, 1990.

[33] Dölling, E., Semiotik und Kognitionswissenschaft, *Zeitschrift für Semiotik, 20, 1-2*, pp. 133-159 (1998).

[34] Smolensky, P., On the proper treatment of connectionism. *Behavioral and Brain Sciences, 11*, pp. 1-74 (1988).

[35] Rosen, R., *Anticipatory systems; philosophical, mathematical, and methodological foundations. IFSR International Series on Systems Science and Engineering – Volume 1.* Oxford: Pergamon Press, 1985.

[36] Goodman, N., *Languages of Arts*. Brighton, Sussex; The Harvester Press, 1968.

[37] Jorna, R.J. & B. van Heusden, Semiotics and information-psychology: a case for semio-psychology. *Theory & Psychology, Vol 8.3,* pp. 755-782 (1998).

[38] Thompson, J. D., *Organizations in Action*. New York: McGraw-Hill, 1967.

[39] Gazendam, W.H.M., *Variety controls variety*. Groningen: Wolters-Noordhoff, 1993.

Global Optimization via SPSA

John L. Maryak and Daniel C. Chin
The Johns Hopkins University, Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, Maryland 20723-6099
john.maryak@jhuapl.edu and daniel.chin@jhuapl.edu
Phone (240) 228-4959 and (240) 228-4956
FAX (240) 228-1093

ABSTRACT

A desire with iterative optimization techniques is that the algorithm reaches the global optimum rather than get stranded at a local optimum value. In this paper, we examine the theoretical and numerical global convergence properties of a certain "gradient free" stochastic approximation algorithm called "SPSA," that has performed well in complex optimization problems. We establish two theorems on the global convergence of SPSA. The first provides conditions under which SPSA will converge in probability to a global optimum using the well-known method of injected noise. The injected noise prevents the algorithm from converging prematurely to a local optimum point. In the second theorem, we show that, under different conditions, "basic" SPSA *without injected noise* can achieve convergence in probability to a global optimum. This occurs because of the noise *effectively* (and automatically) introduced into the algorithm by the special form of the SPSA gradient approximation. This global convergence without injected noise can have important benefits in the setup (tuning) and performance (rate of convergence) of the algorithm. The discussion is supported by numerical studies showing favorable comparisons of SPSA to simulated annealing and genetic algorithms.

KEYWORDS: *Stochastic Optimization, Global Convergence, Stochastic Approximation, Simultaneous Perturbation Stochastic Approximation (SPSA), Recursive Annealing*

1. INTRODUCTION

A problem of great practical importance is the problem of stochastic optimization, which may be stated as the problem of finding a minimum point, $\theta^* \in R^p$, of a real-valued function $L(\theta)$, called the "loss function," that is observed in the presence of noise. Many approaches have been devised for numerous applications over the long history of this problem. A common desire in many applications is that the algorithm reaches the global minimum rather than get stranded at a local minimum value. In this paper, we consider the popular stochastic optimization technique of stochastic approximation (SA), in particular, the form that may be called "gradient-free" SA. This refers to the case where the gradient, $g(\theta) = \partial L(\theta)/\partial\theta$, of the loss function is not readily available or not directly measured (even with noise). This is a common occurrence, for example, in complex systems where the exact functional relationship between the loss function value and the parameters, θ, is not known and the loss function is evaluated by measurements on the system (or by other means, such as simulation). In such cases, one uses instead an approximation to $g(\theta)$ (the well-known form of SA called the Kiefer-Wolfowitz type is an example).

The usual form of this type of SA recursion is:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k), \quad (1)$$

where $\hat{g}_k(\theta)$ is an approximation (at the k^{th} step of the recursion) of the gradient $g(\theta)$, and $\{a_k\}$ is a sequence of positive scalars that decreases to zero (in the standard implementation) and satisfies other properties. This form of SA has been extensively studied, and is known to converge to a local minimum of the loss function under various conditions.

Several authors (e.g., Chin (1994), Gelfand and Mitter (1991), Kushner (1987), and Styblinski and Tang (1990)) have examined the problem of *global* optimization using various forms of gradient-free SA. The usual version of this algorithm is based on using the standard "finite difference" gradient approximation for $\hat{g}_k(\theta)$. It is known that carefully injecting noise into the recursion based on this standard gradient can result in an algorithm that converges (in some sense) to the global minimum. For a discussion of the conditions, results, and proofs, see, e.g., Fang et al. (1997), Gelfand and Mitter (1991), and Kushner (1987). The amplitude of the injected noise is decreased over time (a process called "annealing"), so that the algorithm can finally converge when it reaches the neighborhood of the global minimum point.

A somewhat different version of SA is obtained by using a "simultaneous perturbation" gradient approximation, as described in Spall (1992) for multivariable ($p > 1$) problems. The gradient approximation in simultaneous-perturbation SA (SPSA) is much faster to compute than the finite-difference approximation in multivariable problems. More significantly, using SPSA often results in a recursion that is much more economical, in terms of loss-function evaluations, than the standard version of SA. The loss function evaluations can be the most expensive part of an optimization, especially if computing the loss function requires making measurements on the physical system. Several studies (e.g., Spall (1992), Chin (1997)) have shown SPSA to be very effective in complex optimization problems. A considerable body of theory has been developed for SPSA (Spall (1992), Chin (1997), Dippon and Renz (1997), Spall (2000), and the references therein), but, because of the special form of its gradient approximation, existing theory on global convergence of standard SA algorithms is not directly applicable to SPSA. In Section 2 of this paper, we present a theorem showing that SPSA can achieve global convergence (in probability) by the technique of injecting noise. The "convergence in probability" results of our Theorem 1 (Section 2) and Theorem 2 (Section 3) are standard types of global convergence results. Several authors have shown or discussed global convergence in probability or in distribution (Chiang *et al.* (1987), Gelfand and Mitter (1991), Gelfand and Mitter (1993), Geman and Geman (1984), Fang *et al.* (1997), Hajek (1988), Kushner (1987), Yakowitz *et al.* (2000), and Yin (1999)). Stronger "almost sure" global convergence results seem only to be available by using generally infeasible exhaustive search (Dippon and Fabian (1994)) or random search methods (Yakowitz (1993)), or for cases of optimization in a discrete (θ -) space (Alrefaei and Andradottir (1999)).

The method of injection of noise into the recursions has proven useful, but naturally results in a relative slowing of the rate of convergence of the algorithm (e.g., Yin (1999)) due to the continued injection of noise when the recursion is near a global solution. In addition, the implementation of the extra noise terms adds to the complexity of setting up the algorithm. In Section 3, we present a theorem showing that, under different (more demanding) conditions, the basic version of SPSA can perform as a global optimizer *without* the need for injected noise. Section 4 contains numerical studies demonstrating SPSA's performance compared to two other popular strategies for global optimization, namely, simulated annealing and genetic algorithms; and Section 5 is a summary. The Appendix provides some technical details.

2. SPSA WITH INJECTED NOISE AS A GLOBAL OPTIMIZER

Our first theorem applies to the following algorithm, which is the basic SPSA recursion indicated in equation (1), modified by the addition of extra noise terms:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) + q_k \omega_k , \qquad (2)$$

where $\omega_k \in R^p$ is i.i.d. $N(0,I)$ injected noise, $a_k = a/k$, $q_k^2 = q/k \log\log k$, $a > 0$, $q > 0$, and $\hat{g}_k(\bullet)$ is the "simultaneous perturbation" gradient defined as follows: (3)

$$\hat{g}_k(\theta) \equiv (2c_k \Delta_k)^{-1}[L(\theta + c_k\Delta_k) - L(\theta - c_k\Delta_k) + \varepsilon_k^{(+)} - \varepsilon_k^{(-)}],$$

where $c_k, \varepsilon_k^{(\pm)}$ are scalars, $\Delta_k \in R^p$, and the inverse of a vector is defined to be the vector of inverses. This gradient definition follows that given in Spall (1992). The ε_k terms represent (unknown) additive noise that may contaminate the loss function observation, c_k and Δ_{kl} are parameters of the algorithm, the c_k sequence decreases to zero, and the Δ_{kl} components are chosen randomly according to the conditions in Spall (1992), usually (but not necessarily) from the Bernoulli (± 1) distribution. (Uniformly or normally distributed perturbations are *not* allowed by the regularity conditions.)

In this Section, we will refer to Gelfand and Mitter (1991) as GM91. Our theorem on global convergence of SPSA using injected noise is based on a result in GM91. In order to state the theorem, we need to develop some notation, starting with the definition of a key probability measure, π^η, used in hypothesis H8 below. Define for any $\eta > 0$: $d\pi^\eta(\theta)/d\theta = \exp(-2L(\theta)/\eta^2)/Z^\eta$, where $Z^\eta = \int_{R^p} \exp(-2L(\theta)/\eta^2)d\theta$. Next, define an important constant, C_0, for convergence theory as follows (GM91). For $t \in R$ and $\upsilon_1, \upsilon_2 \in R^p$, let

$$I(t,\upsilon_1,\upsilon_2) = \inf_{\phi} \frac{1}{2}\int_0^t |d\phi(s)/ds + g(\phi(s))|^2 \, ds,$$

where the *inf* is taken over all absolutely continuous functions $\phi: R \to R^p$ such that $\phi(0) = \upsilon_1$ and $\phi(t) = \upsilon_2$, and $|\bullet|$ is the Euclidean norm. Let $V(\upsilon_1,\upsilon_2) = \lim_{t\to\infty} I(t,\upsilon_1,\upsilon_2)$, and $S_0 = \{\theta \,|\, g(\theta) = 0\}$. Then

$$C_0 \equiv \frac{3}{2} \sup_{\upsilon_1,\upsilon_2 \in S_0} (V(\upsilon_1,\upsilon_2) - 2L(\upsilon_2)).$$

We will also need the following definition of *tightness*. If K is a compact subset of R^p and $\{X_k\}$ is a sequence of random p-dimensional vectors, then $\{X_k\}$ is tight in K if $X_0 \in K$ and for any $\varepsilon > 0$, there exists a compact subset $K_\varepsilon \subset R^p$ such that $P(X_k \in K_\varepsilon) > 1 - \varepsilon, \forall k > 0$. Finally, let $\zeta_k^* \equiv \hat{g}_k(\hat{\theta}_k) - g(\hat{\theta}_k)$ and let superscript prime (′) denote transpose.

The following are the hypotheses used in Theorem 1.

H1.. Let $\Delta_k \in R^p$ be a vector of p mutually independent mean-zero random variables $\{\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp}\}'$ such that $\{\Delta_k\}$ is a mutually independent sequence that is also independent of the sequences $\{\hat{\theta}_1,...,\hat{\theta}_{k-1}\}$, $\{\varepsilon_1^{(\pm)},...,\varepsilon_{k-1}^{(\pm)}\}$, and $\{\omega_1,...,\omega_{k-1}\}$, and such that Δ_{ki} is symmetrically distributed about zero, $|\Delta_{ki}| \le \alpha_1 < \infty$ a.s. and $E|\Delta_{ki}^{-2}| \le \alpha_2 < \infty$, a.s. $\forall\, i,k$.

H2.. Let $\varepsilon_k^{(+)}$ and $\varepsilon_k^{(-)}$ represent random measurement noise terms that satisfy $E_k(\varepsilon_k^{(+)} - \varepsilon_k^{(-)}) = 0$ a.s. $\forall k$, where E_k denotes the conditional expectation given $\Im_k \equiv$ the sigma algebra induced by $\{\hat{\theta}_0, \omega_1,..., \omega_{k-1}, \zeta_1^*,...,\zeta_{k-1}^*\}$. The $\{\varepsilon_k^{(\pm)}\}$ sequences are not assumed independent. Assume that $E_k[(\varepsilon_k^{(\pm)})^2] \le \alpha_3 < \infty$ a.s. $\forall k$.

H3. $L(\theta)$ is a thrice continuously differentiable map from R^p into R^1; $L(\theta)$ attains the minimum value of zero; as $|\theta| \to \infty$, we have $L(\theta) \to \infty$ and $|g(\theta)| \to \infty$; $\inf(|g(\theta)|^2 - Lap(L(\theta))) > -\infty$ (*Lap* here is the Laplacian, i.e., the sum of the second derivatives of $L(\theta)$ with respect to each of its components); $L^{(3)}(\theta) \equiv \partial^3 L(\theta)/\partial\theta'\partial\theta'\partial\theta'$ exists continuously with individual elements satisfying $|L^{(3)}_{i_1 i_2 i_3}(\theta)| \le \alpha_5 < \infty$.

H4. The algorithm parameters have the form $a_k = a/k$, $c_k = c/k^\gamma$, for $k = 1,2,...$, where $a, c > 0$, $q/a > C_0$, and $\gamma \in [1/6, 1/2)$.

H5. $[(4p-4)/(4p-3)]^{1/2} < \liminf_{|\theta| \to \infty} (g(\theta)'\theta/(|g(\theta)||\theta|))$.

H6. $E_k(L(\hat{\theta}_k \pm c_k\Delta_k))^2 \le \alpha_4 < \infty$ a.s. $\forall k$.

H7. Let ω_k be an i.i.d. $N(0,I)$ sequence, independent of the sequences $\{\hat{\theta}_1,...,\hat{\theta}_{k-1}\}$, $\{\varepsilon_1^{(\pm)},...,\varepsilon_{k-1}^{(\pm)}\}$, and $\{\Delta_1,...,\Delta_{k-1}\}$.

H8. For any $\eta > 0, Z^\eta < \infty$; π^η has a unique weak limit π as $\eta \to 0$.

H9. There exists a compact subset K of R^p such that $\{\hat{\theta}_k\}$ is tight in K.

Comments:

(a) Assumptions H3, H5, and H8 correspond to assumptions (A1) through (A3) of GM91; assumptions H4 and H9 supply the hypotheses stated in GM91's Theorem 2; and the definitions of a_k and q_k given in equation (2) correspond to those used in GM91. Since we will show that assumption (A4) of GM91 is satisfied by our algorithm, this allows us to use the conclusion of their Theorem 2.

(b) The domain of γ given in H4 is one commonly assumed for convergence results (e.g., Spall (1992)).

We can now state our first theorem as follows:

Theorem 1: Under hypotheses H1 through H9, $\hat{\theta}_k$ converges in probability to the set of global minima of $L(\theta)$.

Proof: See Maryak and Chin (1999), and the remark on convergence in probability in GM91, p. 1003.

3. SPSA WITHOUT INJECTED NOISE AS A GLOBAL OPTIMIZER

As indicated in the introduction above, the injection of noise into an algorithm, while providing for global optimization, introduces some difficulties such as the need for more "tuning" of the extra terms and retarded convergence in the vicinity of the solution, due to the continued addition of noise. This effect on the rate of convergence of an algorithm using injected noise is technically subtle, but may have an important influence on the algorithm's performance. In particular, Yin (1999) shows that an algorithm of the form (2) converges at a rate proportional to $\sqrt{\log\log(k+const)}$, while the nominal local convergence rate for an algorithm *without* injected noise is $k^{1/3}$, i.e., $k^{1/3}(\hat{\theta}_k - \theta^*)$ converges in distribution (Spall (1992)). These rates indicate a significant difference in performance between the two algorithms.

A certain characteristic of the SPSA gradient approximation led us to question whether SPSA needed to use injected noise for global convergence. Although this gradient approximation tends to work very well in an SA recursion, the SPSA gradient, evaluated at any single point in θ-space, tends to be less accurate than the standard finite-difference gradient approximation evaluated at θ. So, one is led to consider whether the *effective* noise introduced (automatically) into the recursion by this inaccuracy is sufficient to provide for global convergence *without* a further injection of additive noise. It turns out that *basic* SPSA (i.e., *without* injected noise) does indeed achieve the same type of global convergence as in Theorem 1, but under a different, and more difficult to check, set of conditions.

In this Section, we designate Kushner (1987) as K87, and Kushner and Yin (1997) as KY97. Here we are working with the basic SPSA algorithm having the same form as equation (1):

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k), \tag{4}$$

where $\hat{g}_k(\bullet)$ is the simultaneous-perturbation approximate gradient defined in Section 2, and now (obviously) no extra noise is injected into the algorithm. For use in the subsequent discussion, it will be convenient to define

$$b_k(\hat{\theta}_k) \equiv E(\hat{g}_k(\hat{\theta}_k) - g(\hat{\theta}_k) \mid \aleph_k), \text{ and } e_k(\hat{\theta}_k) \equiv \hat{g}_k(\hat{\theta}_k) - E(\hat{g}_k(\hat{\theta}_k) \mid \aleph_k),$$

where $\aleph_k$ denotes the σ-algebra generated by $\{\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_k\}$, which allows us to write equation (4) as

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k[g(\hat{\theta}_k) + e_k(\hat{\theta}_k) + b_k(\hat{\theta}_k)]. \tag{5}$$

Another key element in the subsequent discussion is the ordinary differential equation (ODE):

$$\dot{\theta} = g(\theta), \tag{6}$$

which, in Lemma 1 of the Appendix is shown to be the "limit mean" ODE for algorithm (4).

Now we can state our assumptions for Theorem 2, as follows:

J.1 . Let $\Delta_k \in R^p$ be a vector of p mutually independent mean-zero random variables $\{\Delta_{k1}, \Delta_{k2}, ..., \Delta_{kp}\}'$ such that $\{\Delta_k\}$ is a mutually independent sequence and Δ_k is independent of the sequence $\{\hat{\theta}_1, ..., \hat{\theta}_{k-1}\}$, and such that Δ_{ki} is $\forall\, i,k$ symmetrically distributed about zero, $|\Delta_{ki}| \le \alpha_1 < \infty$ a.s. and $E|\Delta_{ki}^{-2}| \le \alpha_2 < \infty$.

J.2 Let $\varepsilon_k^{(+)}$ and $\varepsilon_k^{(-)}$ represent random measurement noise terms that satisfy $E((\varepsilon_k^{(+)} - \varepsilon_k^{(-)}) \mid \aleph_k) = 0$ a.s. $\forall k$. The $\{\varepsilon_k^{(\pm)}\}$ sequences need not be assumed independent. Assume that $E((\varepsilon_k^{(\pm)})^2 \mid \aleph_k) \le \alpha_3 < \infty$ a.s. $\forall k$.

J.3 **(a)**. $L(\theta)$ is thrice continuously differentiable and the individual elements of the third derivative satisfy $|L^{(3)}_{i_1 i_2 i_3}(\theta)| \le \alpha_5 < \infty$.

(b). $|L(\theta)| \to \infty$ as $|\theta| \to \infty$.

J.4 The algorithm parameters satisfy the following: the gains $a_k > 0$, $a_k \to 0$ as $k \to \infty$,a nd $\sum_{k=1}^{\infty} a_k = \infty$. The sequence $\{c_k\}$ is of form $c_k = c/k^{\gamma}$, where $c > 0$ and $\gamma \in [1/6, 1/2)$, and $\sum_{k=0}^{\infty} (a_k / c_k)^2 < \infty$.

J.5 The gradient $g(\theta)$ is bounded and Lipschitz continuous.

J.6 The ODE (6) has a unique solution for each initial condition.

J.7 For the ODE (6), suppose that there exists a finite collection of disjoint compact stable invariant sets (see K87) $K_1, K_2, ..., K_m$, such that $\bigcup_i K_i$c ontains all the limit sets for (6). These sets are interpreted as closed sets containing all local (including global) minima of the loss function.

J.8 For any $\eta > 0, Z^{\eta} < \infty$; π^{η} has a unique weak limit π as $\eta \to 0$ (Z^{η} and π^{η} are defined in Section 2).

J.9 $E|\sum_{i=1}^{k} e_i(\hat{\theta}_i)| < \infty \, \forall k$.

J.10 For any asymptotically stable (in the sense of Liapunov) point, $\bar{\theta}$, of the ODE (6), there exists a neighborhood of the origin in R^p such that the closure, Q_2, of that neighborhood satisfies $\bar{\theta} + Q_2 \equiv \{\bar{\theta} + y : y \in Q_2\} \subset \Theta$, where $\Theta \subset R^p$ denotes the allowable θ-region. There is a neighborhood, Q_1, of the origin in R^p and a real-valued function $H_1(\psi_1, \psi_2)$, continuous in $Q_1 \times Q_2$, whose ψ_1-derivative is continuous on Q_1 for each fixed $\psi_2 \in Q_2$, and such that the following limit holds. For any $\chi, \Delta > 0$, with χ being an integral multiple of Δ, and any functions $\psi_1(\bullet), \psi_2(\bullet)$ taking values in $Q_1 \times Q_2$ and being constant on the intervals $[i\Delta, i\Delta + \Delta), i\Delta < \chi$, we have

$$\int_0^{\chi} H_1(\psi_1(s), \psi_2(s))ds = \lim_{m,n} \sup \frac{\Delta}{m} \log E \exp\left[\sum_{i=0}^{(\chi/\Delta)-1} \psi_1'(i\Delta) \sum_{j=im}^{im+m-1} b_{n+j}(\hat{\theta}_{n+j})\right]. \quad (7)$$

Also, there is a function $H_2(\psi_3)$ that is continuous and differentiable in a small neighborhood of the origin, and such that

$$\int_0^{\chi} H_2(\psi_1(s))ds = \lim_{m,n} \sup \frac{\Delta}{m} \log E \exp\left[\sum_{i=0}^{(\chi/\Delta)-1} \psi_1'(i\Delta) \sum_{j=im}^{im+m-1} e_{n+j}(\hat{\theta}_{n+j})\right]. \quad (8)$$

A bit more notation is needed. Let $T > 0$ be interpreted such that $[0, T]$ is the total time period under consideration in ODE (6). Let

$$\begin{aligned} \bar{H}(\psi_1, \psi_2) &= 0.5[H_1(2\psi_1, \psi_2) + H_2(2\psi_1)] \\ \bar{L}(\beta, \psi_2) &= \sup_{\psi_1}[\psi_1'(\beta - g(\psi_2)) - \bar{H}(\psi_1, \psi_2)]' \end{aligned} \quad (9)$$

and, for $\phi(0) = x \in R^1$, define the function

$$S(T, \phi) = \int_0^T \bar{L}(\dot{\phi}(s), \phi(s))ds,$$

if $\phi(\bullet)$ is a real-valued absolutely-continuous function on $[0, T]$ and to take the value ∞ otherwise. $S(T, \phi)$ is the usual action functional of the theory of large deviations (adapted to our context). Define $t_n \equiv \sum_{i=0}^{n-1} a_i$, and $t_k^n = \sum_{i=0}^{k-1} a_{n+i}$. Define $\{\hat{\theta}_k^n\}$ and $\theta^n(\bullet)$ by

$$\hat{\theta}_0^n = x \in \Theta, \; \hat{\theta}_{k+1}^n = \hat{\theta}_k^n - a_{n+k}\hat{g}_{n+k}(\hat{\theta}_k^n), \text{ and } \theta^n(t) = \hat{\theta}_k^n \text{ for } t \in [t_k^n, t_{k+1}^n).$$

Now we can state the last two assumptions for Theorem 2:

J.11 For each $\delta > 0$ and $i = 1,2,...,m$, there is a ρ-neighborhood of K_i, denoted $N_\rho(K_i)$, and $\delta_\rho > 0, T_\rho < \infty$ such that, for each $x,y \in N_\rho(K_i)$, there is a path, $\phi(\bullet)$, with $\phi(0) = x$, $\phi(T_y) = y$, where $T_y \le T_\rho$ and $S(T_\rho,\phi) \le \delta$.

J.12 There is a sphere, D_1, such that D_1 contains $\bigcup_i K_i$ in its interior, and the trajectories of $\theta^n(\bullet)$ stay in D_1. All paths of ODE (6) starting in D_1 stay in D_1.

Note 1. Assumptions J1, J2, and J3(a) are from Spall (1992), and are used here to characterize the noise terms $b_k(\hat{\theta}_k)$ and $e_k(\hat{\theta}_k)$. Assumption J3(b) is used on page 178 of K87. Assumption J4 expresses standard conditions on the algorithm parameters (see Spall (1992)), and implies hypothesis (A10.2) in KY97, p. 174. Assumptions J5 and J6 correspond to hypothesis (A10.1) in KY97, p. 174. Assumption J7 is from K87, p. 175. Assumption J8 concerns the limiting distribution of $\hat{\theta}_k$. Assumption J9 is used to establish the "mean" criterion for the martingale sequence in Lemma 2. Assumptions J11 and J12 are the "controllability" hypothesis A4.1 and the hypothesis A4.2, respectively, of K87, p. 176.

Note 2. Assumption J10 corresponds to hypotheses (A10.5) and (A10.6) in KY97, pp. 179-181. Although these hypotheses are standard forms for this type of large deviation analysis, it is important to justify their reasonableness. The first part (equation (7), involving noise terms $b_k(\hat{\theta}_k)$) of J10 is justified by the discussion in KY97, p. 174, which notes that the results of their subsection 6.10 are valid if the noise terms (that they denote ξ_n) are bounded. This discussion is applicable to our algorithm since the $b_k(\hat{\theta}_k)$ noise terms were shown by Spall (1992) to be $O(c_k^2)\,(c_k \to 0)$ a.s. The second part (equation (8), involving noise terms $e_k(\hat{\theta}_k)$) is justified by the discussion in KY97, p. 174, which notes that the results in their subsection 6.10 are valid if the noise terms they denote δM_n (corresponding to our noise terms $e_k(\hat{\theta}_k)$) satisfy the martingale difference property that we have established in Lemma 2 of the Appendix.

Now we can state our main theorem:

Theorem 2. Under assumptions J1 through J12, $\hat{\theta}_k$ converges in probability to the set of global minima of $L(\theta)$.

The idea of the proof is as follows (see the Appendix for the details). This theorem follows from results (in a different context) in K87 for an algorithm $\hat{\theta}_{k+1} = \hat{\theta}_k - a_k[g(\hat{\theta}_k) + \zeta_k]$, where ζ_k is i.i.d. Gaussian (injected) noise. In order to prove our Theorem 2, we start by writing the SPSA recursion as $\hat{\theta}_{k+1} = \hat{\theta}_k - a_k[g(\hat{\theta}_k) + \zeta_k^*]$, where $\zeta_k^* \equiv \hat{g}_k(\hat{\theta}_k) - g(\hat{\theta}_k)$ is the "effective noise" introduced by the inaccuracy of the SPSA gradient approximation. So, our algorithm has the same form as that in K87. However, since ζ_k^* is not i.i.d. Gaussian, we cannot use K87's result directly. Instead, we use material in Kushner and Yin (1997) to establish a key "large deviation" result related to our algorithm (4), which allows the result in K87 to be used with ζ_k^* replacing the ζ_k in his algorithm.

4. NUMERICAL STUDIES: SPSA WITHOUT INJECTED NOISE

4.1. Two-Dimensional Problem

A study was done to compare the performance of SPSA to a recently published application of the popular genetic algorithm (GA). The loss function is the well-known Griewank function (see Haataja (1999)) defined for a two-dimensional $\theta = (t_1, t_2)'$, by:

$$L(\theta) = \cos(t_1 - 100)\cos[(t_2 - 100)/\sqrt{2}] - [(t_1 - 100)^2 + (t_2 - 100)^2]/4000 - 1,$$

which has thousands of local minima in the vicinity of a single global minimum at $\theta = (100,100)'$ at which $L(\theta) = 0$. Haataja (1999) describes the application of a GA to this function (actually, to find the *maximum* of $-L(\theta)$) based on noise-free evaluations of $L(\theta)$ (i.e., $\varepsilon_k = 0$). This study achieved a success rate of 66% (see Haataja's Table 1.3, p.16) in 50 independent trials of the GA, using 300 generations and 9000 $L(\theta)$ evaluations in each run of the GA. Haataja's definition of a successful solution is a reported solution where the norm of the solution minus the correct value, θ^*, is less than 0.2, and the value of the loss function at the reported solution is within 0.01 of the correct value of zero. We examined the performance of basic SPSA (without adding injected noise) on this problem, using $a_k = a/(A+k)^\alpha$, with $A = 60$, $a = 100$ and $\alpha = .602$, a slowly decreasing gain sequence of a form that has been used in many applications (see Spall (1998)). For the gradient approximation (equation (3)), we chose each component of Δ_k to be an independent sample from a Bernoulli (± 1) distribution, and $c_k = c/k^\gamma$, with $c = 10$ and $\gamma = .101$. Since we used the exact loss function, the ε_k noise terms were zero. We ran SPSA, allowing 3000 function evaluations in each of 50 runs, and starting the algorithm (each time) at a point randomly chosen in the domain $[-200, 400] \times [-200, 400]$. Haataja's θ-domain was also constrained to lie in a box, but the dimensions of the box were not specified. Hence we chose a domain that is a cube centered at the global minimum, in which there are many local minima of $L(\theta)$ (as seen in Haataja's (1999) Figure 1.1). SPSA successfully located the global minimum in all 50 runs (100% success rate).

4.2. Ten-Dimensional Problem

For a more ambitious test of the global performance of SPSA, we applied SPSA to a loss function given in Example 6 of Styblinski and Tang (1990), which we will designate for convenience as ST90. The loss function is:

$$L(\theta) = (2p)^{-1}\sum_{i=1}^{p} t_i^2 - 4p\prod_{i=1}^{p}\cos(t_i),$$

where $p = 10$ and $\theta = (t_1, \ldots, t_p)'$. This function has the global minimum value of -40 at the origin, and a large number of local minima. As in the two-dimensional study above, we used the exact loss function. Our goal is to compare the performance of SPSA without injected noise with simulated annealing and with a GA.

For the simulated annealing algorithm, we use the results reported in ST90. They used an advanced form of simulated annealing called fast simulated annealing (FSA). According to ST90, FSA has proven to be much more efficient than classical simulated annealing due to using Cauchy (rather than Gaussian) sampling and using a fast (inversely linear in time) cooling scheme. For more details on FSA, see ST90. The results of their application of FSA to the above $L(\theta)$ are given in Table 1 below (FSA values taken from Table 10 of ST90). Table 1 shows the results of 10 independent runs of each algorithm. In each case (each run of each algorithm), the best value of $L(\theta)$ found by the algorithm is shown. In their study, although FSA was allowed to use 50,000 function evaluations for each of the runs, the algorithm showed very limited success in locating the global minimum. It should be noted that the main purpose of the ST90 paper was to examine a relatively new algorithm, stochastic approximation combined with convolution smoothing. This algorithm, which they call SAS, was much more effective than FSA, yielding results between those shown in Table 1 for GA and SPSA.

For the genetic algorithm (GA), we implemented a GA using the popular features of elitism (elite members of the old population pass unchanged into the new population), tournament selection (tournament size = 2), and real-number encoding (see Mitchell (1996), pp. 168, 170, and 157, respectively). After considerable experimentation, we found the following settings for the GA algorithm to provide the best performance on this problem. The population size was 100, the number of elite members (those carried forward unchanged) in each generation was 10, the crossover rate was 0.8, and mutation was accomplished by adding a Gaussian random variable with mean zero and standard deviation 0.01 to each component of the offspring. The original population of 100 (10-dimensional) θ-vectors was created by uniformly randomly generating points in the 10-dimensional hypercube centered at the origin, with edges of length 6 (so, all components had absolute value less than or equal to 3 radians). We constrained all component values in subsequent generations to be less than or equal to 4.5 in absolute value. This worked a bit better than constraining them to be less than 3, since, with the tighter constraints, the GA got stuck at the constraint boundary and could not reach local minima that were just over the boundary. All runs of the GA algorithm reported here used 50,000 evaluations of the loss function. The results of the 10 independent runs of GA are shown in Table 1. Although the algorithm did reasonably well in getting close to the minimum loss value of –40, it only found the global minimum in one of the 10 runs (run #8). In the other nine cases, a few (typically two or four) of the components were trapped in a local minimum (around $\pm pi$ radians), while the rest of the components (approximately) achieved the correct value of zero. Note that the nature of the loss function is such that the value of $L(\theta)$ is very close to an integer (e.g., –39.0 or –38.0) when an even number (e.g., 2 or 4) of components of θ are near $\pm pi$ radians.

We examined the performance of basic SPSA (without adding injected noise), using the algorithm parameters defined in Subsection 4.1 with $A = 60$, $a = 1$, $\alpha = .602$, $c = 2$, and $\gamma = .101$. We started θ at $t_i = 3$ radians, $i = 1,...,p$, resulting in an initial loss function value of -31. This choice of starting point was at the outer boundary of the domain in which we chose initial values for the GA algorithm, and we did not constrain the search space for SPSA as we did for GA (the initialization and search space for FSA were not reported in ST90). We ran 10 Monte Carlo trials (i.e., randomly varying the choices of Δ_k). The results are tabulated in Table 1. The results of these numerical studies show a strong performance of the basic SPSA algorithm in difficult global optimization problems

Table 1. **Best Loss Function Value in Each of 10 Independent Runs of Three Algorithms**

Run	SPSA	GA	FSA
1	–40.0	–38.0	–24.9
2	–40.0	–39.0	–15.5
3	–40.0	–39.0	–29.0
4	–40.0	–38.0	–32.1
5	–40.0	–37.0	–30.2
6	–40.0	–39.0	–30.1
7	–40.0	–38.0	–27.9
8	–40.0	–40.0	–20.9
9	–40.0	–38.0	–28.5
10	–40.0	–39.0	–34.6
Average Value	–40.0	–38.5	–27.4
Number of Function Evaluations	2,500	50,000	50,000

5. SUMMARY

SPSA is an efficient gradient-free SA algorithm that has performed well on a variety of complex optimization problems. We showed in Section 2 that, as with some standard SA algorithms, adding injected noise to the basic SPSA algorithm can result in a global optimizer. More significantly, in Section 3, we showed that, under certain conditions, the basic SPSA recursion can achieve global convergence *without the need for injected noise*. The use of basic SPSA as a global optimizer can ease the implementation of the global optimizer (no need to tune the injected noise) and result in a significantly faster rate of convergence (no extra noise corrupting the algorithm in the vicinity of the solution). In the numerical studies, we found significantly better performance of SPSA as a global optimizer than for the popular simulated annealing and genetic algorithm methods, which are often recommended for global optimization. In particular, in the case of a 10-dimensional optimization parameter (θ), the fast simulated annealing and genetic algorithms generally failed to find the global solution.

APPENDIX (LEMMAS RELATED TO THEOREM 2 AND PROOF OF THEOREM 2)

In this Appendix, we designate Kushner (1987) as K87, and Kushner and Yin (1997) as KY97. Here we are working with the basic SPSA algorithm as defined in equation (4):

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) .$$

We first establish an important preliminary result that is needed in order to apply the results from K87 and KY97 in the proof of Theorem 2.

Lemma 1. The ordinary differential equation (eq. (6) above),

$$\dot{\theta} = g(\theta) ,$$

is the "limit mean ODE" for algorithm (4).

Proof: Examining the definition of limit mean ODE given in KY97, pp. 174 & 138, it is clear that we need to prove that $\frac{1}{m}\sum_{k=n}^{m+n-1}[g(\theta)+e_k(\hat{\theta}_k)+b_k(\hat{\theta}_k)] \to g(\theta)$ w.p. 1 as $m,n \to \infty$. Since Spall

(1992) has shown that $b_k(\hat{\theta}_k) \to 0$ w.p. 1, we can conclude using Cesaro summability that the contribution of the $b_k(\hat{\theta}_k)$ terms to the limit is zero w.p. 1. For the $e_k(\hat{\theta}_k)$ terms, we have by definition that $E[e_k(\hat{\theta}_k)] = 0$; hence, by the law of large numbers, the contribution of the $e_k(\hat{\theta}_k)$ terms to the limit is also zero. Q.E.D.

Our next Lemma relates to Note 2 in Section 3.

Lemma 2. Under assumptions J1, J3(a), and J9, the sequence $\{e_k(\hat{\theta}_k)\}$ is a $\aleph_k$-martingale difference.

Proof: It is sufficient to show that $M_k \equiv \sum_{i=1}^{k} e_k(\hat{\theta}_k)$ is a $\aleph_k$-martingale. Assumption J9 satisfies the first requirement (see KY97, p.68) of the martingale definition, that $E|M_k| < \infty$. For the main requirement, we have for any k:

$$E(M_{k+1} \mid M_k,\ldots,M_1) = E\left[e_{k+1}\left(\hat{\theta}_{k+1}\right) + M_k \mid M_k,\ldots,M_1\right]$$
$$= M_k + E[e_{k+1}(\hat{\theta}_{k+1}) \mid M_k,\ldots,M_1]$$
$$= M_k + E\{[\hat{g}_{k+1}(\hat{\theta}_{k+1}) - E(\hat{g}_{k+1}(\hat{\theta}_{k+1}) \mid \hat{\theta}_{k+1})] \mid M_k\}$$
$$= M_k + E_{\hat{\theta}_{k+1}} E\{[\hat{g}_{k+1}(\hat{\theta}_{k+1}) - E(\hat{g}_{k+1}(\hat{\theta}_{k+1}) \mid \hat{\theta}_{k+1})] \mid M_k, \hat{\theta}_{k+1}\}$$
$$= M_k + E_{\hat{\theta}_{k+1}} E(\hat{g}_{k+1}(\hat{\theta}_{k+1}) \mid M_k, \hat{\theta}_{k+1}) - E_{\hat{\theta}_{k+1}} E[E(\hat{g}_{k+1}(\hat{\theta}_{k+1}) \mid M_k, \hat{\theta}_{k+1}] = M_k,$$

where $E_{\hat{\theta}_{k+1}}$ denotes expectation conditional on $\hat{\theta}_{k+1}$, and all equalities concerning conditional expectations are w.p. 1. Q.E.D.

A key step in the proof of our main result (Theorem 2 below) is establishing the following "large deviation" result (Lemma 3). Let B_x be a set of continuous functions on $[0,T]$ taking values in Θ and with initial value x. Let B_x^0 denote the interior of B_x, and $\overline{B}_x$ denote the closure.

Lemma 3. Under assumptions J4, J5, J6, and J10, we have

$$-\inf_{\phi \in B_x^0} S(T,\phi) \le \liminf_n \log P_x^n\{\theta^n(\bullet) \in B_x\}$$
$$\le \limsup_n \log P_x^n\{\theta^n(\bullet) \in B_x\} \le -\inf_{\phi \in \overline{B}_x} S(T,\phi), \quad (9)$$

where P_x^n denotes the probability under the condition that $\theta^n(0) = x$.

Proof: This result is adapted from Theorem 10.4 in KY97, p. 181. Note that our assumption J10 is a modified form of their assumptions (A10.5) and (A10.6), using "equals" signs rather than inequalities. The two-sided inequality in (9) follows from J10 by an argument analogous to the proof of KY87's Theorem 10.1 (p. 178), which uses an "equality" assumption ((A10.4), p. 174) to arrive at a two-sided large deviation result analogous to (9) above. Q.E.D.

We restate our main theorem:

Theorem 2: Under hypotheses J1 through J12, $\hat{\theta}_k$ converges in probability to the set of global minima of $L(\theta)$.

Proof: This result follows from a discussion in K87. Theorem 2 of K87, (p.177) describes probabilities involving expected times for the SA algorithm (system (1.1) of K87) to transition from one K_i to another. The SA algorithm he uses can be written in our notation as $\hat{\theta}_{k+1} = \hat{\theta}_k - a_k[g(\hat{\theta}_k) + \zeta_k]$, where ζ_k is i.i.d. Gaussian (injected) noise. The K87 Theorem 2 uses the i.i.d. Gaussian assumption only to arrive at a large deviation result exactly analogous to our Lemma 3. The subsequent results in K87 are based on this large deviation result. Recall that the SPSA algorithm without injected noise can be written in the form $\hat{\theta}_{k+1} = \hat{\theta}_k - a_k[g(\hat{\theta}_k) + \zeta_k^*]$. Since we have established Lemma 3 for SPSA, the results of K87 hold for the SPSA algorithm with its "effective" noise $\{\zeta_k^*\}$ replacing the $\{\xi_k\}$ sequence used in K87. In particular, K87's discussion (pp. 178, 179) of his Theorem 2 is applicable to *our* Theorem 2 context (SPSA without injected noise), which corresponds to K87's "potential case." Note that our formulation corresponds to the K87 setup where $b(x,\xi) = \bar{b}(x)$ in his notation, which, by the comment in K87, p. 179, means that his discussion is applicable to his system (1.1) and hence to our setup. In his discussion on p. 179, K87 indicates that the difference between the measure of X_n (which corresponds to our $\hat{\theta}_k$) and the invariant measure (which we have denoted π^{η}) converges asymptotically ($n,k \to \infty, \eta \to 0$) to the zero measure weakly. This means that, in the limit as $k \to \infty$, $\hat{\theta}_k$ is equivalent to π in the same sense as in Theorem 2 of Gelfand and Mitter (1991), and the desired convergence in probability follows as in Theorem 1 above. Q.E.D.

ACKNOWLEDGEMENT

This work was partially supported by the JHU/APL Independent Research and Development Program and U.S. Navy contract N00024-98-D-8124.

REFERENCES

[1] Alrefaei, M.H. and Andradottir, S. (1999), "A Simulated Annealing Algorithm with Constant Temperature for Discrete Stochastic Optimization," *Management Science*, 45, pp. 748-764.

[2] Chaing T-S., Hwang, C-R., and Sheu, S-J. (1987), "Diffusion for Global Optimization in R^n," *SIAM J. Control Optim.*, 25, pp. 737-753.

[3] Chin, D.C. (1997), "Comparative Study of Stochastic Algorithms for System Optimization Based on Gradient Approximations," *IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics*, 27, pp. 244-249.

[4] Chin, D.C. (1994), "A More Efficient Global Optimization Algorithm Based on Styblinski and Tang," *Neural Networks*, 7, pp. 573-574.

[5] Dippon, J. and Fabian, V. (1994), "Stochastic Approximation of Global Minimum Points," *J. Statistical Planning and Inference*, 41, pp. 327-347.

[6] Dippon, J. and Renz, J. (1997), "Weighted Means in Stochastic Approximation of Minima," *SIAM J. Control Optim.,* 35, pp. 1811-1827.

[7] Fang, H., Gong, G., and Qian, M. (1997), "Annealing of Iterative Stochastic Schemes," *SIAM J. Control Optim.*, 35, pp.1886-1907.

[8] Gelfand, S.B. and Mitter, S.K. (1991), "Recursive Stochastic Algorithms for Global Optimization in R^d," *SIAM J. Control Optim.*, 29, pp. 999-1018.

[9] Gelfand, S.B. and Mitter, S.K. (1993), "Metropolis-Type Annealing Algorithms for Global Optimization in R^d," *SIAM J. Control Optim.*, 31, pp. 110-131.

[10] Geman S. and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-6, pp. 721-741.

[11] Haataja, J. (1999), "Using Genetic Algorithms for Optimization: Technology Transfer in Action," Chapter 1, pp. 3 – 22, in *Evolutionary Algorithms in Engineering and Computer Science,* Edited by K. Miettinen, M.M. Makela, P. Neittaanmaki, and J. Periaux, Wiley, Chichester.

[12] Hajek, (1988), "Cooling Schedules for Optimal Annealing," *Mathematics of Operations Research*, 13, pp. 311-329.

[13] Kushner, H.J. and Yin, G.G. (1997), *Stochastic Approximation Algorithms and Applications*, Springer, New York.

[14] Kushner, H.J. (1987), "Asymptotic Global Behavior for Stochastic Approximation and Diffusions with Slowly Decreasing Noise Effects: Global Minimization Via Monte Carlo," *SIAM J. Appl. Math.*, 47, pp. 169-185.

[15] Maryak, J.L. and Chin, D.C. (1999), "Efficient Global Optimization Using SPSA," *Proc. Amer. Control Conf.*, San Diego, June 2-4, pp. 890-894.

[16] Mitchell, M. (1996), *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass.

[17] Spall, J.C. (2000), "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method," *IEEE Trans. Automat. Control,* 45, pp. 1839-1853.

[18] Spall, J.C. (1998), "Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization," *IEEE Trans. Aerospace and Electronic Systems,* **34**, pp. 817-823.

[19] Spall, J.C. (1992), "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Trans. Automat. Control*, 37, pp. 332-341.

[20] Styblinski, M.A. and Tang, T-S. (1990), "Experiments in Nonconvex Optimization: Stochastic Approximation with Function Smoothing and Simulated Annealing," *Neural Networks*, 3, pp. 467-483.

[21] Yakowitz, S. (1993), "A Globally Convergent Stochastic Approximation," *SIAM J. Control Optim.* 31 pp. 30-40.

[22] Yakowitz, S., L'Ecuyer, P., and Vazquez-Abad, F. (2000), "Global Stochastic Optimization with Low-Dispersion Point Sets," *Operations Research*, 48, pp. 939-950.

[23] Yin, G. (1999), "Rates of Convergence for a Class of Global Stochastic Optimization Algorithms," *SIAM J. Optim.*, 10, pp. 99-120.

PART I
RESEARCH PAPERS

General Discussion Panel 1

What is the role of Ontology in Performance Evaluation?

Ontology in Performance Evaluation

Edward Dawidowicz
Command and Control Directorate
US Army, CECOM, RDEC

The criteria for measurement of performance of intelligent systems allow a qualitative comparison of one system against another. The performance measurement allows us not only to judge the achieved level of success or failure, but also to evaluate cost effectiveness of a particular design. However, the performance criteria alone is not sufficient to give an objective evaluation of the system. A thorough approach requires the consideration of methods and devices for performance evaluation. The need for this thorough approach requires us to take a look at the ontology of performance of intelligent systems.

.

Before we attempt to define methods of performance measurements we need to reexamine what are the distinctive elements which make a system intelligent and how do they relate to each other. These elements require scrutiny individually and as whole [1] as well as the mechanisms, by means of which they interact.

The questions that are worthwhile to examine are those that reflect processes at a particular level of system functioning associated with organizational echelons. An intelligent system is goal driven. The goals are both internal and external. The system has to be able to decompose goals into subgoals giving an emergence of hierarchy of goals. The system representation, also hierarchical, plays an important part in developing the ontology and requires a distinct attention [3].

Another tempting aspect for discussion is the topic of ontology of self-referentiality of intelligent system [2]. Particularly considering the new and different ways in which intelligent systems conceptually forms and specifies representation of objects as a particular manifestation of self-referentiality. That indeed requires special attention when formulating ontology.

A few question arise as we ponder the meaning of the role of ontology in performance evaluation:

1. What is the purpose of ontology and how does it helps us to find the answers we are looking for?
2. What are the framework, development methodologies and life-cycle maintenance?
3. Can it provide an objective representation of performance and associative measurement devices and methodologies?
4. Should the ontology reflect the system functioning and collaboration with other systems?
5. An intelligent system has a user who interacts with it. A user maybe another system or a human. Does that mean that we need two ontologies, one for the user and the other for a system under evaluation?
6. And finally when we are addressing complex systems, and an intelligent system is a complex system, should we not consider ontologies of many hierarchical levels [3]?

References:

1. John H. Holland, "Emergence: From Chaos to Order", Helix Books
2. Roger Penrose, "Shadows of the Mind", Oxford university press, 1994
3. Alexander Meystel, James Albus, " Intelligent Systems: Architecture, Design and Control, Wiley, NY, 2002.

Using Ontologies to Evaluate Knowledge-based Systems

Michael Gruninger

NIST

Many systems have hidden assumptions about their domain that must be rendered explicit if we are to identify the principles on which they are based. Verified ontologies provide one way of making these assumptions explicit. A verified ontology consists of a specification of a class of mathematical structures together with a proof of two fundamental properties:

- **Satisfiability**: every structure in the class is a model of the ontology's axioms;
- **Axiomatizability**: every model of the ontology's axioms is isomorphic to some structure in the class.

Strictly speaking, we only need to show that a model exists in order to demonstrate that an ontology is satisfiable. However, in the axiomatization of ontologies, we need a complete characterization of the possible models. For example, if we are considering the domain of activities, occurrences, and timepoints, to show that a theory is satisfiable, we need only specify an occurrence of an activity that together with the axioms are satisfied by some structure. The problem with this approach is that we run the risk of having demonstrated satisfiability only for some restricted class of activities. For example, a theory of activities that supports scheduling may be shown to be consistent by constructing a satisfying interpretation, but the interpretation may require that resources cannot be shared by multiple activities or it may require all activities to be deterministic. Although such a model may be adequate for such activities, it would in no way be general enough for our purposes; we would want a comprehensive theory of activities that explicitly characterize the classes of activities, timepoints, objects, and other assumptions that are guaranteed to be satisfied by the specified structures.

When implementing knowledge-based systems, we are faced with the additional challenge that almost no existing software application has an explicitly axiomatized ontology. However, we can model a software application *as if* it were an inference system with an axiomatized ontology, and use this ontology to predict the set of sentences that the inference system decides to be satisfiable. This is the *Ontological Stance*, and is analogous to Dennett's intentional stance, which is the strategy of interpreting the behavior of an entity by treating it as if it were a rational agent who performs activities in accordance with some set of intentional constraints.

Using the ontological stance, we can define capabilities of knowledge-based systems and explain why certain techniques fail when extended to new domains. In particular, we can characterize the knowledge used in a given domain, and how this knowledge influences a particular reasoning task. Ontologies support a semantic assumption-based approach to tractable reasoning -- rather than identify syntactic classes of theories, we can reason about the assumptions that the ontology entails.

An example of this is the CardWorld Ontology that axiomatizes shape-based object recognition in scenes consisting of 2D surfaces with occlusion. The ontology allows characterization of the complexity of finding a model of an image together with the axioms of the ontology. In particular, tractable subclasses can be defined by

- Assumptions on images (e.g. accidental alignments)
- Assumptions on scenes (e.g. layered surfaces)
- Assumptions on depiction (e.g. errors in edge detection)

Towards an Ontology of Performance

Line Pouchard
Computer Science and Mathematics Division
Oak Ridge National Laboratory

Specifying the concepts for an ontology of performance advances PERMIS's project to define metrics of intelligence. The multiple factors briefly mentioned below highlight a few aspects of the complexity and depth of concepts and issues to be considered. It is therefore useful to keep in mind questions that arise when developing an ontology:

- What will be the purpose, use and scope of this ontology? Defining a consensus on terminology, defining machine-readable schemas, a practical and/or theoretical purpose, interoperability between systems, interface between experts in related fields are good examples.
- Are domain specific ontologies advantageous?
- Who will be the users and stakeholders of this ontology? (human communication, computerized interaction, human-machine interaction?)
- How and by whom will it be developed? Which language is chosen for representing the ontologies?

The performance of a distributed system results from trade-offs between properly weighed factors that include computational costs and communication overhead on the one hand, and computational and communication benefits on the other. Computational benefits include the number of transactions per seconds or milliseconds, the throughput of Input/Output of the system as a whole, and its response time.[i] Computational costs depend on the load of each component, e.g. number and complexity of processes utilized in performing the task at hand, and the consumption of computational resources. Communication costs may depend on the number of components, their model of interaction and the general architecture of a system.

Computational costs and benefits depend on a compromise between amount and type of resources consumed by the system. For instance, in large distributed systems (order of dozen of nodes distributed across the country) analyzing data-intensive scientific data (order of petabytes) such as the type of applications for which Grid computing[ii] is designed, the computational costs also vary depending on the type of architecture. The type of architecture here means distributed data and centralized processing, versus distributed data and distributed processing. If data needs to be shipped to a central very powerful computer (super-computer of the type pioneered by Cray research)[iii], as current implementation of Grid applications require, the load on computational resources will be large and so will be computational costs. The load in term will affect response time that depends on hardware performance as well as the algorithms driving the hardware.

In the case of distributed data and distributed processing, data is no longer shipped to a central location. Instead, a software code or a software component is moved to a remote computational resource close to data storage.

Communication costs may depend on the number of components interacting with each other, the availability and cost of network bandwidth, and the architecture of communication. Architecture of communication here may mean the protocol of communication [iv] (for example contract-net protocol in multi-agent systems, asynchronous communication such as message passing in parallel systems). Communication overhead may also include factors related to the structure of a system, whether a hierarchical structure or a web-based structure are used. In the hierarchical structure, children interact with its parents and children; in a web-based structure, any component may a priori interact with any other.

Affecting performance, one also finds factors related to the individual performance of a component independently of the performance of the system as a whole. The knowledge base of a component and the complexity of its rules for reasoning upon input are such factors. Thus several levels of measuring performance in a system may be envisioned.

What Role Can Ontologies Play to Improve the Performance of Intelligent Systems?

Craig Schlenoff
Intelligent Systems Division
National Institute of Standards and Technology

There are at least three overarching questions that one may ask regarding the role of ontologies in performance evaluation. They are:

1. How can an ontology play a role in evaluating intelligent systems?
2. How can an ontology within an intelligent system be evaluated?
3. How can ontologies play a role in helping an intelligent system perform at a higher level of performance?

This write-up focuses on this third question.

When most people think about the role of ontologies in performance evaluation, they often ask question such as:

- What will be the purpose of the ontology?
- How formal of an ontology does one need?
- How does one create such ontologies?

To attempt to answer these questions, one can imagine introducing an ontology (or set of ontologies) into an autonomous vehicular system's knowledge base. Let's assume that the ontology consisted of various objects that the vehicle expected to encounter in its environment, as well as the important characteristics of each object. This ontology would help to answer question such as:

- Based upon the data we get from our sensors, what are the objects we perceive in the environment at the given time, with appropriate levels of confidence?
- What characteristics of those objects do I need to be most concerned about?

With the introduction of ontologies representing factors that effect the motion of objects (such as an ontology representing the "rules of the road", motion limitation of certain vehicles, the network of roads in the environment), we could answer questions such as:

- Where do we expect an object to be at a time in the future?
- What is our risk of colliding with the object assume the motion patterns do not change?

Assuming the ontology has clearly defined semantics, we can use the ontology to unambiguously exchange information among different autonomous vehicles (or between a vehicle and a human) that are working together to jointly accomplish a goal.

With the introduction of an ontology of actions that an autonomous vehicle is able to perform, we could answer questions such as:

- How can the overall goal of the vehicle be decomposed into actionable items?
- What are the appropriate actions in a given situation?

Based on these ontologies, if the entire environment around the vehicle at any given time could be completed modeled, we could evaluate the planner by answering questions such as:

- Has the planner identified all possible plans?
- Does the chosen plan truly accomplish the stated goal?

Therefore, I contend that ontologies can play a role in improving the performance of almost every part of autonomous vehicular systems, and most likely in intelligent systems as a whole.

However, that is not to say that there aren't various technical challenges that have to be addressed before ontologies can be used in this context. Some of the questions that need to be address include:

- How can ontologies be linked to other types of representations, including sensor data?
- Can ontologies respond quickly enough to be useful in a real-time environment?
- Are ontologies reusable in intelligent systems, and if so, what is the best mechanism to share them with other?
- How can one evaluate the performance of the ontology?

Role of Ontologies in Performance Evaluation

Lawrence A. Welsch, Ph.D. – LWelsch@nist.gov

There are many possible roles that an ontology may have evaluating the performance of intelligent systems. I only list some of the more obvious roles that I see

Correctness of Results

Solutions to problems and answers to questions from the system being evaluated may not always be what is expected. For example, an expected answer might be *4-wheel vehicle*. The system being evaluated might respond go-cart, car, truck, vehicle, tank, ice cream bar, or any one of a number of answers. Car and go-cart are probably correct answers. Truck may or may not be correct, but is certainly a vehicle. Tank is a vehicle, but not wheeled. Ice cream bar would not even be close. An ontology would help make those decisions rapidly.

Was the Explanation Correct

When performing a test of intelligence, the tester may want to see an explanation of how the answer was derived. An ontology can be used to

1. determine whether or not the knowledge used in an explanation is consistent (may be indeterminate within the time allotted);
2. determine whether or not the steps are logical; and
3. whether or not a better explanation exists.

Why did the System being Evaluated do X?

Sometimes judgements of performance are made based upon the behavior of the system. If we understand why the system is behaving as it is, then there is not need to ask for an explanation. However if we do not understand then we need to ask why. An ontology can be used to understand the behavior of a system being evaluated and if it doesn't then to ask why. If the system being evaluated has made an incorrect assumption in it's explanation then an ontology could be used to probe the rationale for the poor assumption.

How Optimal was the Answer?

Different systems have different quantities of knowledge, represented in different ways. Given absolute or total knowledge of the problem domain, then how good was the answer, solution, explanation given by the system compared with the best possible answer, solution, explanation for the problem/question.

Ontologies OF and FOR Intelligent Systems

A. Meystel
Drexel University, NIST

1. Constructing an ontology of *Intelligent System (IS)* is a nontrivial issue since its (IS) existence has not been demonstrated, and the term is used as a metaphor. It raises a number of additional questions. Not the last among them is whether existential problems can be legitimately formulated for the domain of imaginary worlds. The answer is probably an affirmative one, since imaginary worlds do exist (at least in the subsystem of planning) and thus generate corresponding ontologies. Another question is linked with processes of *reflexia* that emerge within intelligent systems. This really opens a can of worms since individual and group processes of reflection are based upon the property of self-reference – not a trivial subject for the theory of IS ontology. This set of problems is difficult but is pleasurable because it requires surveying a multicultural (at least, multidisciplinary) domain including psychological and linguistic references as well as engineering ones.

2. Constructing an ontology for *Intelligent System (IS)* is inescapable because its (IS) functioning dwells upon input, output, and intermediate (e. g. interface, etc.) ontologies. The need in multi-resolutional ontologies is looming, although the ontology community temporarily (eternally?) resists making this one of the regular issues of research. The difficulty of the latter subject is in the fact, that despite of the unity of the multi-resolutional ontology of the IS, it needs consistency checks to be conducted at each level. Nevertheless, the community seems to be well prepared for solving this issue, too.

PART I
RESEARCH PAPERS

Reception

RoboCup: Humanoids as a Testbed for Measuring Performance

Minoru Asada
Adaptive Machine Systems
Osaka University, JAPAN
August 13th, 2002
PerMES-02@NIST

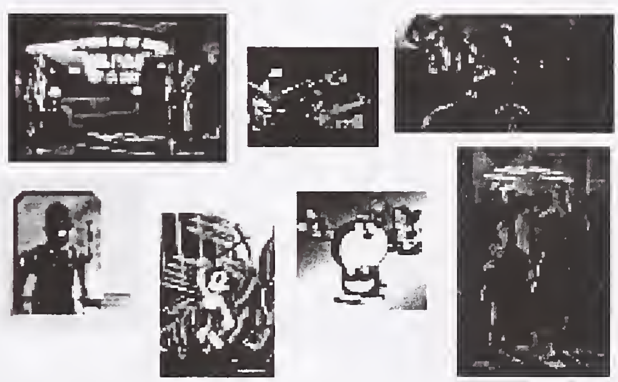
RoboCup 2002 Fukuoa/Busan

- The Biggest RoboCup since 1997
  - 1004 participants, 188 teams from 30 nations around world, and about 1000 media people.
  - About 120,000 visitors during one press day and four open public days
- The first humanoid robot league
  - 13 teams from 6 nations
- ROBOTREX (Robot Trade & Exhibitions)
  - 50 companies, universities, and instittutes

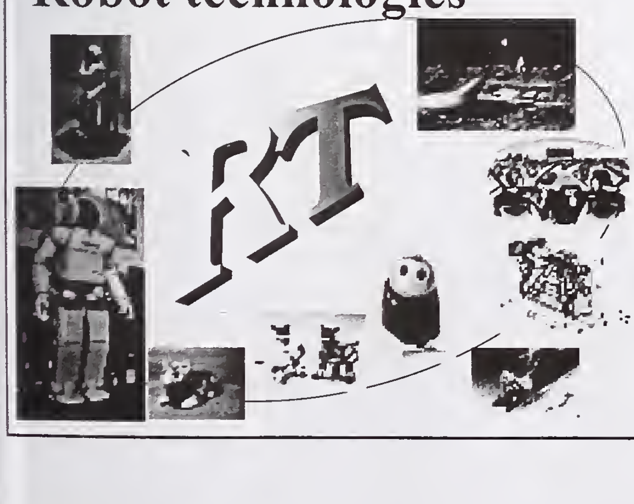
Outline of my talk

- Robot technology
  - What's robot
- RoboCup
  - Purpose, Current State, and Issues.
- Technological issues towards finial goal: Humanoid league
  - Levels of autonomy
  - One leg standing, walk, PK, and free style

Roboto Heros and Heroines

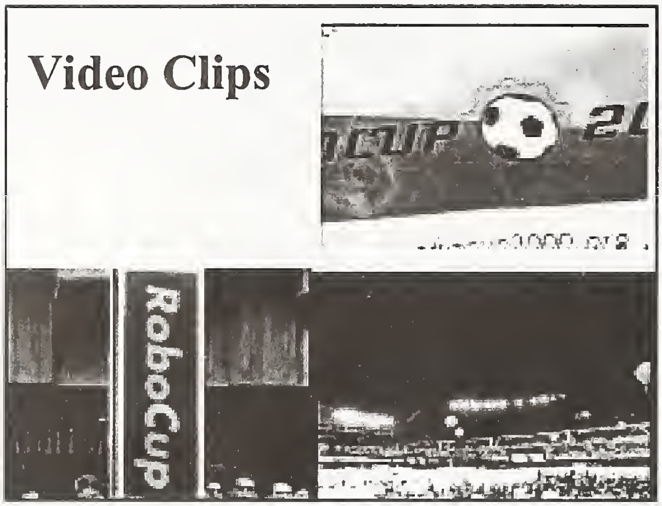


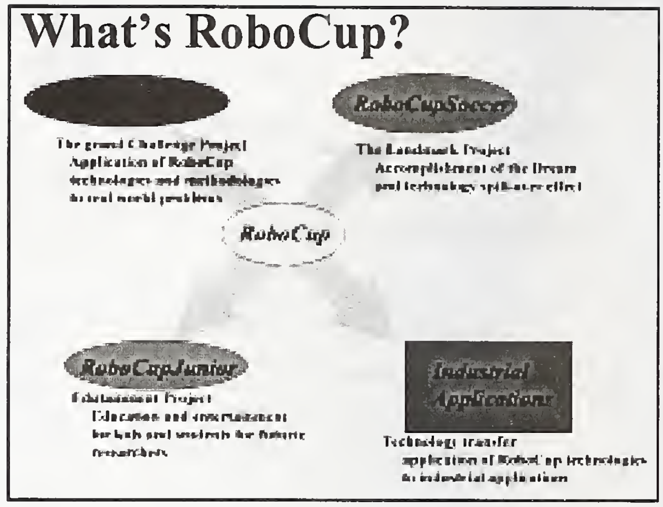
Robot technologies

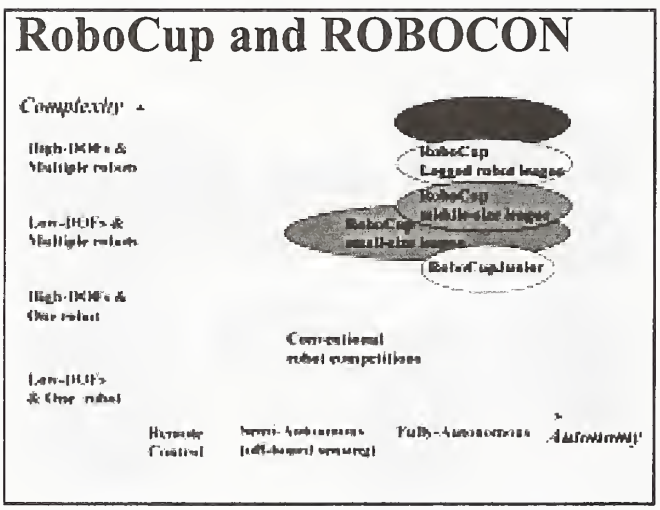


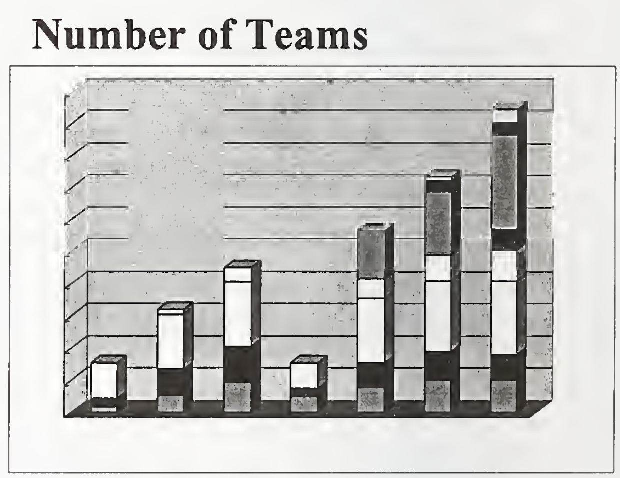
What's robot? (I)

- Perception
  - See, Listen, Smell,...
  - Somatosensory
- Cognition and Decision
  - What's that? Do that and then,...
- Action
  - Walk, Run, Kick, Grasp,..


Video Clips
RoboCup

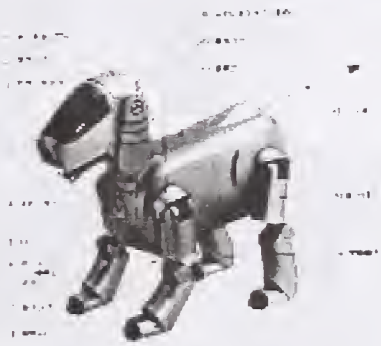

What's RoboCup?
RoboCupSoccer
RoboCup
RoboCupJunior
Industrial Applications


RoboCup and ROBOCON
Complexity
RoboCupJunior
Conventional robot competitions

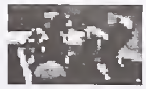

Number of Teams

What's robot? (II)

- energy
- sensor
- computer
- actuator
- mechanism



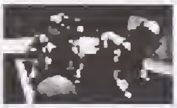
History of robotics (I)



- The term robot comes from the Czech word robota, meaning drudgery or slave-like labor, and from the old church Slavonic robota, meaning servitude.
- It was first used to describe fabricated workers in a fictional 1920s play by Czech author Karel Capek called Rossum's Universal Robots.
- Three laws of robtics by Asimov
- Special purpose or general pupose

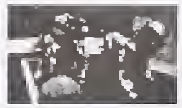
→Healing and communication

History of robotics (II)



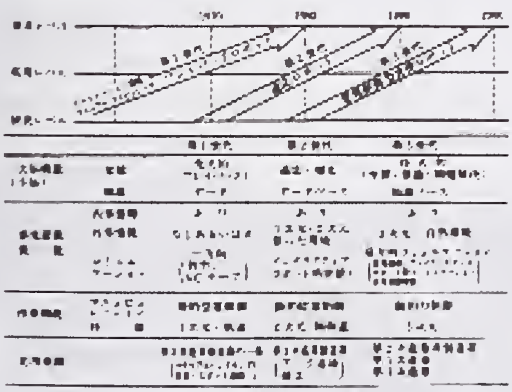
- wind-up dolls

History of robotics (III)



- Industrial Robot
- Numerical Control
- Sensor based Control
- Adaptation/Learning
- Intelligent robots

Generation of Robots



What's RoboCup?

An attempt to foster intelligent robotics research by providing a standard problem

Research Issues in RoboCup (I)

- Mechanical design for individual robots
- Robust Sensing, especially, vision (object discrimination and tracking)
- Self-localization and map building
- Control Architecture
- Communication

Research Issues in RoboCup (II)

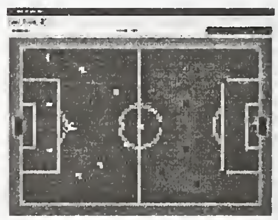
- Multi-agent systems in general
- Behavior learning for complex tasks
- Combining reactive and modeling approaches
- Real-time recognition, reasoning, planning, and action execution in a dynamic environment
- Cross modal association (Sensor fusion)
- Strategy acquisition
- Cognitive modeling in general

Divisions of RoboCup

- RoboCupSoccer
  - Simulation: Coach, Visualization
  - Real robot: Small, Middle, Legged, and Humanoid
- RoboCupRescue
  - Simulation and real robot
- RoboCupJunior
  - Soccer, Dance, and Rescue

Soccer Simulation League

- Low cost, Stamina model, 11 v.s. 11, limited perception, broadcasting



- Secondary Domain →RoboCup-Rescue

Soccer Simulation League

- Teamwork
- On-line learning
- Coach competition
- Visualization

Real Robot Leagues

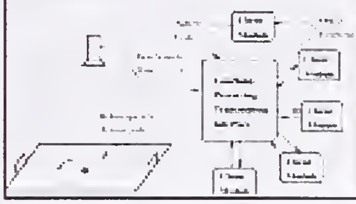
- Small Size league: A table tennis table, an orange golf ball, and global vision..
- Middle Size league: 3X3 table tennis tables, an official soccer ball, and local vision..

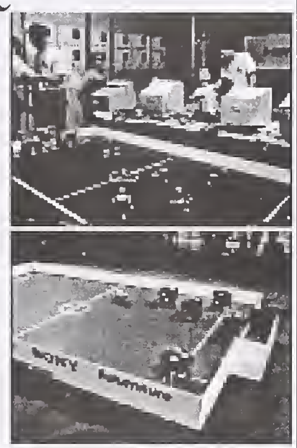
Real Robot Leagues (cntd.)

- Legged league: Sony AIBO Type robots, 4 on 4.
- Humanoid league: Four classed according to the size. One leg standing, walk, PK, and free style

Small-size league

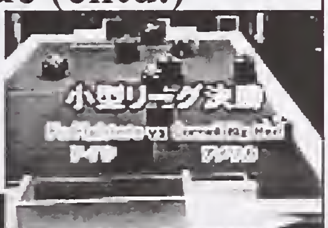
- 1997~ Global vision:
- Perception: Sharing global information, but reliable and real-time detection of multi mobile robots and ball.





Small-size league (cntd.)

- 2000, field: wooden → fabric
- 2002, enlargement of the filed to encourage the team plays.



Middle-size league

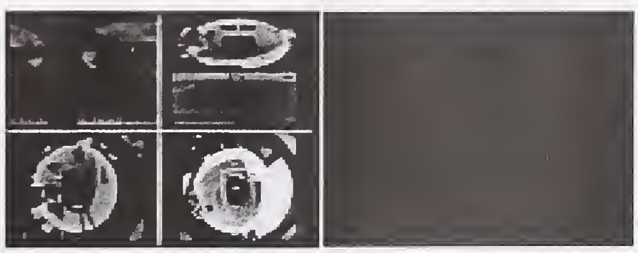
- Fully distributed system, but centralized control is OK!
- Evolution from individual behavior to team plays.

Middle-size league (cntd.)

- Global reconstruction by LRFs (C. S. Freiburg, Germany)

Middle-size league (cntd.)

- 1997~ 5 on 5 Footsul-4 ball
- 2000~ 4 on 4 Footsul-5 (official ball)
- Omni-directional vision, Reactive behavior, and social ones.



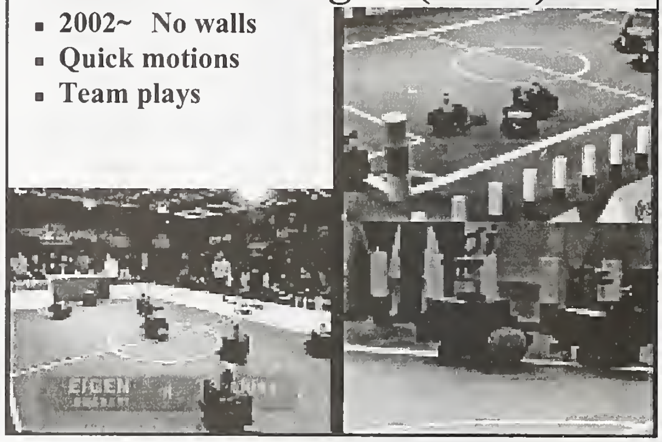
Middle-size league (cntd.)

- Holonomic Vehicles: Omni-directional movement



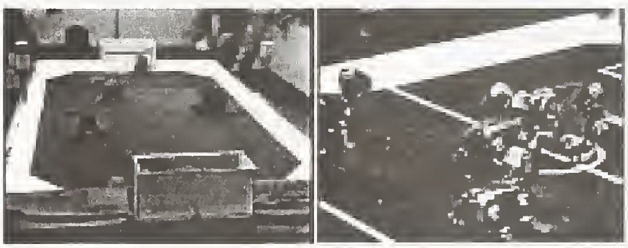

Middle-size league (cntd.)

- 2002~ No walls
- Quick motions
- Team plays



Legged league

- Programming competition based on the same platform
- 1998 exhibision (Osaka U. CMU Paris-VI)
- 1999~ Official league (2m 3m 3 on 3)



Legged league (cntd.)

- Various kinds of behaviors Ball handling
- Teamwork social behavior based on vocal communication

Legged league (cntd.)

- 2001~ New platform
- 2002~ 3m 4m 4 on 4 wireless communication

Humanoid league

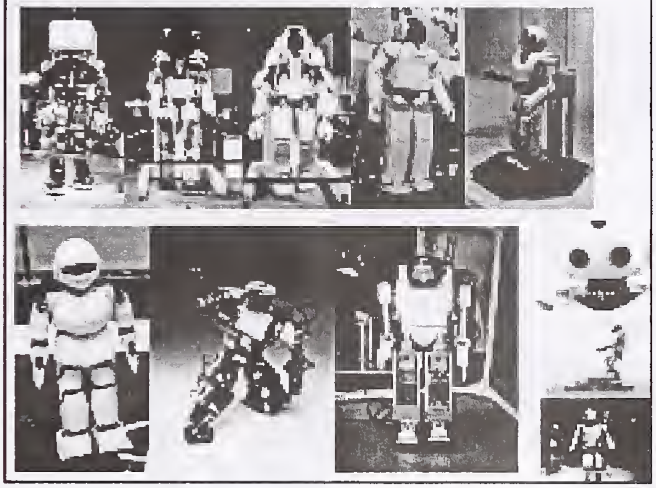
- 2002 the first humanoid robot soccer.
- 4 kinds of size 40cm 80cm 120cm 180cm
- Perfomance factor towards fully autonomous humanoid robot: Platform, power supply from outside, remote brain, human control
- Japan (5), Sweden (3), Singapore (2), New Zealand, Australia, Denmark

Why humanoid robot?

- One extreme application: test-bed for brain science.

- Current application: entertainment, pet robots.

- Another extreme application: practical use in our daily life??? → HRP by government.



Perfomance Factor

- We would like to trigger developments towards fully autonomous self-build humanoid robots.
- Therefore we took so-called *performance factors* for the different dimensions with regard to autonomy.
  1. external power cord
  2. computer outside robot
  3. remote control
  4. Platform
- Each were to be 1.2 and if more then one is applicable then they are multiplied (1.2, 1.44, 1.728, 2.0736).

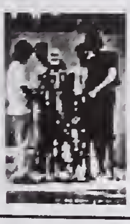
Perfomance Factor (cntd.)

- These factors were either used
  1. as penalty factor in the walking the time was multiplied by them or
  2. as handicap (in penalty kicking the score was divided by them).

- They are working quite well (with regard to the above stated intention) and will certainly prefer the more autonomous robots but will also allow for semi-autonomous ones if their performance is much better then that of the autonomous ones. No changes needed.

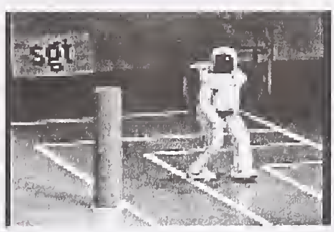
Challenges: stand on one leg

- This is definitely no problem for most of the humanoid robots or it shouldn't be one while it is one for humans! It is a wonderful entry if the audience is also involved in this. It was done in Fukuoka by asking everybody in the audience to perform this challenge together with the robots.



Challenges: walking

- A round trip of humanoid walking along the way of five times its height.
- Every touch of a human during the walking gives a penalty which is linearly increasing: 20 sec/1st touch, 40 sec/2nd touch, 60 sec/3rd touch etc.
- Champion:
  - Nagara (Japan)
  - 81,64, and 61 seconds
  - 3.29 (p/f: 1.0)
- Second:
  - Robo-Erectus (SG)
  - 209, 109, and 183 secs.
  - 4.932 (p/f: 1.2)

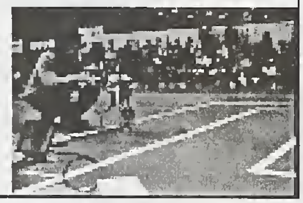


Challenges: Penalty kick

- Total behavior coordination with walking, one leg standing, kicking, and balancing.
- The physical height of the striking robot was used to determine the distance between ball and striker while the measurements of the goals were only available for the two categories (40 cm and 80 cm height).

Challenges: Penalty kick (cntd.)

- First, to give the striker a realistic chance we introduced a 5 sec latency after the starting whistle before the goalie may start to walk towards the ball to reduce the angle which could be used to score a goal.
- Second, the line of the goal area was used a strict demarcation line to avoid the collision.
- The was so light that it often went astray due to small uneven parts in the field.



Challenge: free style

- Honda Asimo's example performance and digest from humanoid league.

Humanoid league: photos

Humanoid league: issues

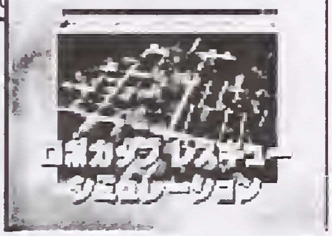
- Performance factor: what values and how to apply?
- Stand on one leg: difficult to decide real time sensor feedback or open loop. Introduction of disturbance to check it.
- PK: from PK to 2 on 2!
- Free style: A test bed for humanoid research in general

RoboCupRescue Simulation

RoboCupRescue Simulation (cntd.)

500 x 500 m region in Nagata Ward, Kobe City

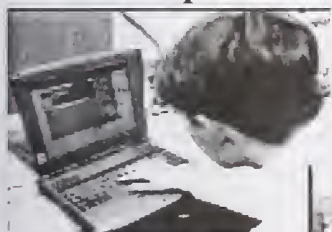
Multi-layered human interface with information filt

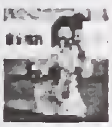


RoboCupRescue Real robot

- Evaluation of cooperation deployed in three stage rescue situation by NIST

RoboCupJunior





Future Issues

- Road Map towards the final goal: set up Milestones.
- Humanoid league: regulations

Time table

- Spring, 2003: regional events: Japan Open, German Open, US Open
- July, 2003: The seventh RoboCup at Padoa, Italy
- July, 2004: The eighth RoboCup at Lisbon, Portugal

Acknowledgement

- RoboCup Federation,
- NPO RoboCup Japanese Committee
- Humanoid Chair: Prof. Dr. Thomas Christaller

http://www.robocup.org/

RoboCup

PART I
RESEARCH PAPERS

Plenary Lecture 2 - 1

Evolving Solutions that are Competitive with Humans
D. Fogel, Natural Selections, Inc.

Evolving Solutions that are Competitive with Humans

David B. Fogel
Natural Selection, Inc.
3333 N. Torrey Pines Ct., Suite 200
La Jolla, CA 92037
dfogel@natural-selection.com

Extended Abstract

Ever since the advent of the modern digital computer, we've tried to generate machines that are as intelligent as humans. Indeed, this was the primary goal of early artificial intelligence (AI) researchers. The "general problem solver" served as one example of efforts along this line, and it's lack of success was disappointing to many in the AI community. One of the primary deficiencies of early, and even modern, efforts in AI has been a lack of a definition of the intelligence that is sought. There is no accepted definition of intelligence, let alone "artificial intelligence." In place of careful definitions, the Turing Test evolved as a surrogate criterion for judging intelligence.

The famous Turing Test involves an interrogator asking questions of a woman and man via a teletype machine, with the goal of correctly identifying the woman. The woman presumably gives truthful answers to questions, while the man may lie. Turing proposed replacing the man with a computer and suggested that if a computer can fool the interrogator into believing that it is the woman as often as a man can fool the interrogator then the machine will be said to have passed the test. Interestingly, Turing never claimed that passing the test meant that such a machine would be "intelligent," an issue that he described as being "too meaningless to deserve discussion."

In retrospect, the Turing Test is no more a test for intelligence than it is a test for femininity. If a man can fool an interrogator into believing that he is the woman, that does not make the man a woman. Similarly, just because a computer might fool an interrogator into believing that it is intelligent does not make the computer intelligent. Nevertheless, the Turing Test has had a profound impact on efforts to simulate behaviors that we associate with intelligence, mainly as we observe them in ourselves. Unfortunately, over time, the impact was mainly to narrow the focus of AI to simply generate programs that could compete with humans in specialized areas, such as chess. The mechanism for generating the required behavior became irrelevant, as all that was of importance was the end result. The culmination of this process is Deep Blue, a very fast machine that can beat Garry Kasparov in chess, but is no more "intelligent" than a calculator, or a hammer.

Rather than begin from the perspective of the Turing Test, an alternative perspective begins with the concept of decision making. For an organism to be intelligent, it must make decisions. It is pointless to speak of the intelligence of something that does not make decisions. A decision can be defined to arise when available resources are allocated. Note that a range of possible decisions, possible allocations, must be available otherwise there really is no decision at all. Logically, decision making requires a goal, for decision making in the absence of a goal is pointless. Thus we must inquire as to where goals come from.

In natural systems, the primary goal instilled in all living organisms is survival. Those organisms that do not possess this goal may be "successful," but are uninteresting from an evolutionary perspective. Thus behaviors can be judged in how well they support this ultimate goal, and subgoals that underlie it. More generally then, intelligence can be defined as "the ability for a system to adapt its behavior to meet its goals in a range of environments." This capability of intelligent decision making can be observed strikingly in the evolving phyletic lines of organisms (the reader is free to choose which ones), such as frogs and insects in which cryptic coloration, poison, reliable signaling, and mimicking have all been invented to meet the primary goal of "avoiding being someone else's lunch." No single individual invented any of these "tricks," rather the intelligent organism in these cases is the evolving line of individuals.

Taking this cue from nature, it is reasonable to assess the intelligence capability of a machine that evolves solutions to problems in a manner similar to that of evolving phyletic lines in the natural environment. From the perspective of performance comparison, many efforts in evolutionary computation have been measured in light of human capabilities. For example, L. Fogel (Artificial Intelligence Through Simulated Evolution, Wiley, 1966) compared the ability of graduate students and an evolutionary program operating on finite state machines to predict sequences of symbols. The results showed that the evolutionary program was competitive or slightly more capable than its human competitors. In Germany, in the mid-1960s, H.-P. Schwefel used an evolutionary algorithm to create a new design for a flashing nozzle, which exceeded the capabilities of the previous human design. There are many other relevant results in the literature.

More recently, the author and a colleague (Kumar Chellapilla) investigated the ability for an evolutionary algorithm to learn to play checkers at a level that is commensurate with human experts, without relying on human expertise about checkers. Instead, neural networks were used to evaluate candidate board positions based only on the inputs found in the number, location, and types of pieces on the board. Furthermore, the neural networks were not told which games were won, lost, or drawn. Only an overall point value was assessed to each neural network, which signified the total value earned over a series of games.

Starting from randomly weighted connections, a population of neural networks evolved, using random variation of the weights of each neural network and a selective mechanism to eliminate poor-scoring networks, over 100 generations to be competitive with "Class

B" human players on the Internet. With some modifications of the input design to the neural networks, which allowed a recognition that the game is played on a two-dimensional board, and 840 generations, the best-evolved neural network (called Blondie24) was able to compete with human experts and finished in the top 500 of over 120,000 people on the Internet site, www.zone.com. More details on this effort can be found in D. Fogel's book, Blondie24: Playing at the Edge of AI, Morgan Kaufmann, 2002.

The results indicate that evolution is a suitable mechanism for creating intelligent behavior in machines, and that it can learn to generate behavior that is competitive with human experts even without relying on human expertise. As computer hardware increases in speed according to Moore's Law, it is important to recall that this acceleration in speed is not sufficient to generate intelligent machines. The software that this hardware will execute is critically important. The results described here and presented in the plenary lecture indicate one step toward creating intelligent machines that may someday possess an ability to adapt their behavior, to meet their goals, in a range of environments that is commensurate with our own abilities.

PART I
RESEARCH PAPERS

2M1 - Uncertainty of Representation I

Evaluation Methods for Human-System Performance of Intelligent Systems

Dr. Jean Scholtz
Visualization and Usability Group
Information Access Division
Information Technology Laboratory
National Institute of Standards and Technology

Abstract

Intelligent systems are becoming more and more of a reality but with the exception of very special purpose systems, completely autonomous systems are not yet the norm. In reality, we need to have humans who monitor the systems, intervening when necessary. As systems increase in intelligence, the goal for human-in-the-loop activities should not be to eliminate the human, but rather to create a human-system partnership with greater capabilities than the individual components. We currently view intelligent systems and the operators or supervisors of these systems as separate components and conduct evaluations in the same vein. For intelligent systems to become more useful and acceptable, we need to consider the "system" as a synergistic composition of software behaviors, possibly embedded in a physical component such as a robot, and the human interacting with this virtual or physical component. Our objective is to design this team interaction in such as way that the intelligence of the team is greater than the intelligence of any one of the parts.

Keywords: *Human-robot interaction, situational awareness, human-computer interaction, evaluation methodologies, intelligent systems..*

1. Introduction

In our work we are concerned with intelligent systems embodied in hardware (robots). Human-robot interaction is fundamentally different from typical human-computer interaction (HCI) in several dimensions. [8] notes that HRI differs from HCI and Human-machine Interaction (HMI) because it concerns systems which have complex, dynamic control systems, exhibit autonomy and cognition, and which operate in changing, real-world environments. In addition, differences occur in the types of interactions (interaction roles), the physical nature of robots, the number of systems a user may be called to interaction with simultaneously, and the environment in which the interactions occur.

The interaction roles of supervisor, operator, and peer are defined in [17]. Upon further consideration we have subdivided two of these roles resulting in five different interaction roles: supervisor, operator, mechanic, teammate, and peer. The supervisory role involves monitoring the intelligent system and seeing that any interventions that are needed are handed off to the proper individual. We have subdivided the original operator role into an operator role and a mechanic role. An operator is needed to work "inside" the robot; adjusting various parameters in the robot's control mechanism to modify abnormal behavior; to change a given behavior to a more appropriate one; or to take over and tele-operate the robot. The mechanic interaction is undertaken when a human needs to adjust physical components of the robot, such as the camera or various mechanical mechanisms. The peer role has been divided into a teammate role and a bystander role. The teammate role implies the same relationship between humans and robots as it does in human-human interactions. Teammates of intelligent systems can interact at an "implementation level." The commands a teammate can give to a robot should not change the nature of the plan or mission but allows adjustments due to the dynamics of a particular situation. A bystander does not explicitly interact with a robot but needs some model of robot behavior as the bystander will be in the same physical space as the robot and needs to co-exist.

The second dimension is the physical nature of mobile robots. Robots need some awareness of the physical world in which they move. As robots move about in the real world, they build up a "world model" [2]. The robot's model needs to be conveyed to the human in order to understand decisions made by the robot as the model may not correspond exactly to reality due to the limitations of the robot's sensors and processing algorithms.

A third dimension is the dynamic nature of the robot platform and its effect on performance and capabilities. In typical human-computer interactions the assumption is that the computer is working and that behavior does not change over time. In assessing user interactions with the internet, we are starting to question this assumption as the workload and the time delays at any particular time can affect what the user does and how satisfied the user is with the experience. The fact that robot capabilities can change implies that functionality at time 1 may not be available at time 2 and this has to be factored into the human-robot interaction.

In typical human-computer interactions the cognitive state of the human has been largely ignored. The notion of affective computing [15] is just starting to appear in commercial products. Human computer interaction specialists in such domains as air traffic control [12], safety critical systems, and military systems have been concerned with these issues for some time and have attempted to designed user interfaces that are usable in these conditions. The current trend is to design interfaces that detect the status of the user and adjust accordingly. Consideration of environmental conditions and the effect on the users is a necessity in HRI.

In human-robot systems several of the interaction roles (supervisor, teammate, bystander) may need to interact with a number of systems simultaneously. These systems may be operating completely independently or they may be functioning as a team. Moreover, several types of interactions could be occurring at the same time; for example, a supervisor might be overseeing a robot which is also interacting with a teammate. Typical HCI considers 1:1 situations; one user to one computer. In HRI, we have the possibility of a 1:N situation where one person is interacting with N robots and N:1 where a number of people are interacting with the same robot.

The autonomous nature of intelligent systems has been a subject of discussion for some time in the human factors world [13]. This changes the user's role from being in control to monitoring and intervening when necessary. This introduces the concept of "being out-of-the-loop" and raises the issue of how to alert the user to an exception and how to bring the user up to speed to quickly and effectively intervene.

2. Approach

Traditional HCI takes a user-centered approach [14] while others in the automation field have taken a system-centric approach [3]. We are taking an information-centric view. It is necessary to determine what information the user needs to understand what the intelligent system is doing and when intervention is necessary, and what information is needed to make any intervention as effective as possible. This understanding is basic to the design of a user interface that is able to present the appropriate information to the user. Intervention requires one more component: a language for the user and the intelligent system to use in resolving the problem. The final part of the problem is actually carrying out the intervention, assessing the situation correctly and giving advice or performing a necessary action for the intelligent system.

We propose six different issues in evaluation that must be considered to evaluate the overall human-intelligent system interaction:

1. Is the necessary information for the human to be able to determine that an intervention is needed present?
2. Is the information presented in an appropriate form?
3. Is the interaction language efficient for both the human and the intelligent system?
4. Are interactions handled efficiently and effectively - both from the user and system perspective?
5. Does the interaction architecture scale to multiple platforms and interactions?
6. Does the interaction architecture support evolution of platforms?

The first four issues are relevant to all intelligent systems. If we are concerned with supporting 1:N and N:1 interactions, we must evaluate the scalability of the interaction architecture. If we are interested in using the architecture over a period of time, we must consider how the evolving behaviors of new intelligent systems will be supported.

Usability evaluations of desktop software products use three metrics: effectiveness, efficiency, and user satisfaction [10]. Due to the dynamic nature of intelligent systems separating the evaluation into two pieces, getting the proper information to the user and the actual performance of the user/system in the interaction, produces a finer granularity of understanding. Users may have all the information they need but the interaction can fail for other reasons. Likewise, the interaction may be successful without users having the proper information. By separating the evaluation into these pieces, we reduce the risk of counting these cases in the results.

These evaluation questions cannot presently be answered in a general sense. Our approach is to narrow both the domain and the role of the human interaction and systematically explore the space. After we have explored a number of roles and domains of interaction, we will examine the results to determine if there are commonalties that can be expressed as guidelines for interaction guidelines.

3. Evaluation Methodologies

The six issues listed above are evaluated using different types of evaluations. In this section we discuss the types of evaluations appropriate for each issue.

3.1. Information Presence and Presentation

To determine if the necessary information is presented – and in the correct form – we are customizing a situational awareness assessment methodology. Situational awareness [6] is the knowledge of what is going on around you. The implication in this definition is that you understand what information is important to attend to in order to acquire situational awareness. Consider your drive home in the evening. As your drive down the freeway and urban streets there is much information you could attend to. You most likely do not notice if someone has painted their house a new color but you definitely notice if a car parked in front of that house starts to pull out in your path.

Level One of situational awareness (SA) is the basic perception of information in your surroundings. For example, in driving did you notice the cars to the left, right, front and rear of your vehicle? Failures to perceive information can result as short comings of a system or they can be due to a user's cognitive failures. In studies of situational awareness in pilots, 76% of SA [11] errors were traced to problems in perception of needed information. Level Two of situation awareness is the ability to comprehend or to integrate multiple pieces of information and determine the relevance to the goals the user wants to achieve. A person achieves the third level of situational awareness if she is able to forecast future situation events and dynamics based on her perception and comprehension of the present situation.

The most common way to measure situational awareness is by direct experimentation using queries [5]. The task is frozen, questions are asked to determine the user's situational assessment at the time, then the task is resumed. The Situation Awareness Global Assessment Technique (SAGAT) tool was developed as a measurement instrument for this methodology 7]. The SAGAT tool uses a goal-directed task analysis to construct a list of the situational awareness requirements for an entire domain or for particular goals and sub-goals. Then it is necessary to construct the query in such a way that the operator's response is minimized. For example, if a user were being queried about the status of a particular robot, the query might present the robot by location rather than replying on the user to recall a name or to understand a description. The various options for status could be presented as choices rather than relying on the user to formulate a response that might not include all the variables desired. SAGAT queries are constructed to include measures of all three levels of situation awareness. Queries related to placement of nearby vehicles in a driving domain would measure level one situation awareness. Queries that ask about vehicle activity, such as cars that have just switched lanes or increased or decreased speed would address level two. Level three, prediction, would be measured by queries such as the likelihood that the car in front of you will move into the far left lane. This could be determined by the observation of a turn signal or the previous pattern of behaviors of lane switching by that automobile.

3.2. Interaction Performance

Information presence and presentation evaluations are user-centric. Interaction performance evaluations need to take into account the performance of both the human and the intelligent system. We are concerned with measuring the ability of the user to formulate the correct interaction and the system to understand and carry this out. This type of evaluation can be conducted as a typical HCI evaluation [4]. To elaborate, a set of tasks are constructed and explained to the user. The user is then directed to use the interface to accomplish each task. The training and the expertise of the user can confound this evaluation. In general, the users should be chosen from the representative population of users and given the same amount of training as those users would be given. If there are a number of diverse users, then different classes of users should be identified and between five and eight users from each class should be used in the evaluation. From the user perspective the metrics should reflect the number of tasks that the user is able to successfully complete, the time for each task, and a user satisfaction measure that can be obtained using a standard questionnaire [16].

The system performance is also factored into this metric. The effectiveness measure has two components – the user executing the correct interaction and the system responding correctly to this interaction. Likewise the efficiency metric would be the sum of the user time for the interaction to be specified and the system time for it to be carried out.

Currently we have not considered the notion of mixed initiative interaction; that is, the intelligent system notices that an interaction by the user is needed and notifies her of this. This would require ensuring that these notifications are seen and understood by the user (the situation awareness measurement) and that the correct interaction is selected and carried out (the interaction performance portion).

3.3. Support for Scalability and Evolution

Support for 1:N and N:1 interactions should be evaluated using the information presence and presentation, and performance methods. The information to be displayed in each case and the presentation of that information needs to be determined and evaluated using a situational awareness assessment. The performance measures need to ensure that the user can identify the appropriate interaction for the appropriate platform within the appropriate time. This will be critical if a number of heterogeneous platforms are being used. In the case of multiple people interacting with the system simultaneously, it will be interesting to determine how to display that information and what effect this will have on the interaction of any given role. For example, how will the operator behavior change if a number of bystanders are present when she needs to teleoperate a robot to get it into a building?

Evaluating the interaction support for evolution is more difficult. Intelligent systems will evolve and be capable of undertaking more tasks successfully and communicating with users at higher levels of abstraction. This will definitely necessitate re-examining the interaction language, but the information that is needed has to be reconsidered as well. A more robust level of autonomy might be supported as effectively using more abstractions in the first level of information presented.

3.4 Evaluation Methodologies for Different Roles

The five roles defined here, supervisor, operator, mechanic, teammate, and bystander, clearly have different information requirements and different interactions. Is it feasible to use the same type of evaluation to measure the performance of all roles?

We expect that the supervisor, operator, and mechanic will have access to a specialized display of information from the intelligent system. This display may be on a workstation, laptop or a small handheld device. However, this display will give the appropriate situational awareness to the users. Teammates may not have access to such a device and bystanders certainly will not. A different methodology is needed in these situations. This will be discussed in section 5.

All of the interaction roles will have access to some subset of the interaction vocabulary. A particular interaction may be selected using a typical command language, keyboard input, voice, or pen-based types of interactions. The action might even be initiated by using "physical" manipulation. For example, moving in front of a robot and touching a sensor on the robot to cause an action to occur would be an example of "physical" manipulation. The same performance evaluations will be used in all roles. However, in cases where no specialized display is available, the challenge will be to make the interaction choices known to the user.

4. Case Study One: Developing the Situational Awareness Assessment Tool

In our current work, we have narrowed the domain to a mobile, robotic platform that is given the task of driving from one place to another in a complex urban environment. The human-robot interaction role is that of the supervisor, similar to a driving coach with a student behind the wheel. As the capabilities of the student driver increase, the driving coach should be able to pay less attention and only give guidance when she detects the possibility of a problem. However, our driving coach will be remote. For the first part of the work we are investigating the first two questions: what information is necessary for the human to decide that intervention is needed and what is the appropriate presentation of this information?

The first step is to determine what information is needed by the user. In the driving domain this task is easier than in most because of the amount of information available. There are numerous studies on driving [15] and our own experience that we will turn to for the initial design of the user interface.

We currently characterize the information using four categories: static environment; dynamic environment; platform information, and task information. The static environment consists of information in the environment that does not change or at least changes very infrequently. In the driving domain, this would be the location of cross streets and intersections; the type of roadway; whether there are stop signs, stop lights, or other traffic controls. Dynamic environment information examples are the amount of traffic present, the pedestrians traffic if the driving environment is urban at the time, and the status of the traffic light. Examples of vehicle information are the speed of the vehicle, the amount of fuel left, and the condition of the vehicle such as non-working turn indicators. Task information is the knowledge of the destination; the current distance to the destination; how far to the next decision point.

For given situations, the information needs change. For example, approaching a green light, drivers should look for different hazards than when approaching a red light.

Once the information for selected situations is determined and the user interface is designed, the awareness assessment tool is constructed. As explained in the earlier section, this is accomplished by using a simulation and freezing the simulation at a certain point. The simulation screen is blanked out and the user is directed to answer a series of questions to determine what her situational awareness is at this time. The queries that are constructed are the most important aspect of the assessment methodology. Again, expert elicitation in some form is used to obtain this information. This can be done by observations of performance, verbal protocols, interviews, or questionnaires. The results can be combined and later verified by a number of subject experts. In the driving domain, we are utilizing a tutorial used to teach driving [16]. The tutorial presents a number of situations and the student is asked to identify potential hazards or take action to prevent accidents. The information given is the front view, the rear and side view mirrors and the instrument panel. The accompanying instructor manual identifies the information that students needed to have identified. We will use this information to construct our queries.

The presentation of the queries should be done so that the user can quickly answer. The users should not be asked to recall information that is not relevant to the situation. For example, asking a user to designate if there are cars to the left, right, front (within 2 car lengths) and back (within 2 car lengths) of his car on a multilane highway is fine. Requesting the number of cars for most other situations is not fine.

The analysis will be done for each situation that we present and for the different information classifications that we have identified. We will use situations from highway driving, urban driving, and illustrating normal conditions as well as hazardous conditions.

We intend to use our user interface and the situational awareness assessment results as a baseline. These will be made public. Others interested in this particular domain could either construct a new user interface to display the same information we have identified and compare their results. Alternatively, different information could be displayed in the user interface and the results compared with a baseline.

Once we have completed our information presence and presentation evaluation we will proceed to the performance evaluation. We intend to also look at scalability of the user interface from one to multiple robotic platforms.

5. Case Study Two: Examining User Mental Models in the Bystander Role

We are also working at the opposite end of the spectrum and looking at evaluation methodologies where no specialized visual user interface is present. We have designed an experiment to examine the effects of consistency and expectedness of behavior on bystanders' abilities to construct a mental model of the robot's capabilities. We are using a Sony Aibo ™ [1] for this experiment and as the robot has a dog-like appearance we have designed behaviors that one would normally expect of a dog (playing with a ball, sitting) and others (singing, dancing) that would not be expected.

[1] * The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology.

Observations of the users as they interact with the robot will be recorded and users will be asked after the experiment to identify what robot behaviors will result from various interactions on their part. Results from the first round of this study will be available this fall.

6. Conclusions

We have defined five different interaction roles for humans and intelligent robotic systems. We have also defined six issues that constitute "performance of intelligent systems." We have outlined the evaluation methodologies that can assess these measures of performance. We are currently conducting evaluation experiments for two types of interaction roles in two domains: the supervisory role in a driving domain and a bystander role in a social interaction domain. Our future plans are to use the framework suggested by roles and domains to systematically explore evaluation methodologies for human-robot performance. Our tools and results will be made publicly available as our research progresses.

7. Acknowledgements

This work is funded by DARPA, contract number E710. The author would like to thank Dr. James Albus and Elena Messina for their support and encouragement in including the human element in their research programs. Brian Antonishek, NIST and Siavosh Bahrami, University of California, Irvine are working on the experiments discussed in this paper.

References

1. AAA Foundation for Traffic Safety. 2002. driver-ZED.
2. Albus, J. 1997. 4-D/RCS: A Reference model Architecture for Demo III, NISTIR 5994, Gaithersburg, MD. March.
3. Brown, M. and Leveson, N. 1998. Modeling Controller Tasks for Safety Analysis. Presented at the Workshop on Human Error and System Development, Seattle, April.
4. Dumas, J. S. & Redish, J. C. 1993. A practical guide to usability testing. Norwood, NJ: Ablex Publishing.
5. Endsley, M. R., 2000. Direct Measurement of Situation Awareness: Validity and Use of SAGAT in (Eds) Mica R. Endsley and Daniel J. Garland . Situation Awareness Analysis and Measurement, Lawrence Erlbaum Associates: Mahwah, New Jersey.
6. Endsley, M. R., 2000. Theoretical Underpinnings of Situation Awareness: A Critical Review, in (Eds) Mica R. Endsley and Daniel J. Garland . Situation Awareness Analysis and Measurement. Lawrence Erlbaum Associates, Publishers: Mahwah, NJ
7. Endsley, M. R. 1988. Design and evaluation for situation awareness enhancement. In Proceedings of the Human Factors Society 32nd Annual meeting (Vol.

1, pp 97-1010 Santa Monica, CA: Human Factors Society.

8. Fong, T., Thorpe, C. and Bauer, C. 2001. Collaboration, Dialogue, and Human-robot Interaction, 10th International Symposium of Robotics Research, November, Lorne, Victoria, Australia.
9. Gugerty, L. 1997. Situation Awareness during Driving: Explicit and Implicit Knowledge in Dynamic Spatial Memory. Journal of Experimental Psychology: Applied. Vol. 3 (1), 42-66.
10. ISO 9241-11. 1998. Ergonomic requirements for office work with visual display terminals (VDT)s part 11 Guidance on usability.
11. Jones, D.G. and Endsley, M. R. 1996. Sources of situation awareness errors in aviation. Aviation, Space and Environmental Medicine 67(6), pp. 507-512.
12. Leveson, N. de Villepin, M., Daouk, M. , Bellingham, J., Srinivasan, J., Neogi, N., Bachelder, E., Pilon, N., and Flynn, G. 2001. A Safety and Human-Centered Approach to Developing New Air Traffic Management Tools. ATM 2001, Albuquerque NM, December
13. Leveson, N. and Palmer, E. 1997. Designing Automation to Reduce Operator Errors In the Proceedings of Systems, Man, and Cybernetics Conference, Oct.
14. Norman, D. 1986. Cognitive Engineering in Donald Norman and Stephen Draper (Eds.) User-centered design:new perspectives on human-computer interaction, Erlbaum Associates: Hillsdale, N.J, 31-62.
15. Picard, R. (1997), Affective Computing, MIT Press, Cambridge.
16. The Software Usability Measurement Inventory. (SUMI) Accessed July 12, 2002. http://www.ucc.ie/hfrg/questionnaires/sumi/
17. Scholtz, J. (2002) Creating Synergistic CyberForces in Alan C. Schultz and Lynne E. Parker (Eds.), Multi-Robot Systems: From Swarms to Intelligent Automata. Kluwer.

Lifelike Robotic Collaboration requires Lifelike Information Integration

Ron Cottam, Willy Ranson and Roger Vounckx
The Evolutionary Processing Group, IMEC Brussels
Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels
ron@cottam.be

Introduction

This paper is directed towards the definition of a systemic characteristic suitable for the "intelligent" core of a cooperative robot. A major issue in developing cooperative systems is the nature and degree of their autonomy. It is rare (rather, impossible!) that a communicated set of instructions for carrying out a task will be exhaustively complete, and with the passage of time the instruction set and the task's requirements often diverge. Neither complete dependence on control by a human "master", nor complete autonomy from control is suitable or desirable within such a context, but it is as yet unclear how a satisfactory context-dependent intermediate *modus* can be developed which is conducive to cooperation while not stifling any pre-existent or nascent capability for creative problem-solving. We believe that detailed examination of a number of more general aspects of system operation should predate attempts at defining performance measures for "intelligent" systems *per se*.

We wish to address four specific aspects of the performance and visualization of large information-processing networks. Firstly (1), the character of information transport through the networked connections of large systems; secondly (2), the way in which establishment of a hierarchical structure can alleviate some of the resulting problems; thirdly (3), the relationship between rationality and emotion within such a scheme; and fourthly (4), the manner in which information is integrated and visualized in human "thought" – which brings us right back to our first chosen systemic aspect (1).

Our starting point is the recognition that our environment cannot be completely described in any detailed manner by using a closed formal system of rationality. The representations we use for parts of our surroundings are all to some extent approximate in ways which relate to the varying nature of their interactions at different scales. Deviations from exact correspondence between descriptions of an entity at its different scales reside in the inter-scalar interfaces, where interactions are naturally complex and predictability is limited. As a simple example of this difficulty we can take a Boolean AND gate with 2 inputs and 1 output. Conversion between the 4 possible input states and 2 possible output states is controlled by the logical rules which correspond to the pre-defined gate function, but even so the gate's operation is irreversible because information is lost in the course of its operation. Reversible state compression demands the retention of *all* independent information, but the only way an AND gate can be made reversible is by recourse to non-local memory… more of this later. In the meantime we should simply note that wherever there is cross-scale information transport we can expect problems in the application of closed formal rationality.

1. Large Systems

Looking into the heart of a system, we often describe its pathways and their meeting points by the simple picture of a network of interlinked lines and nodes. An example is the ball-and-stick models which are used to describe molecules in chemistry. The balls represent not only entities, but also communicational nodes; the sticks represent major communicational pathways. If all of these pathways are similarly simply specified, whether as globally existing or globally absent, then the system is relatively easy to describe, and it can be referred to as being minimally complex. If, however, all of the pathways are individually specified, then the description is of necessity far more complex. We must also decide *very* clearly *where* we are looking at things from in painting our picture, as in reality we only have *one* point of view at one point in time.

A system can be described from an *external* platform as a set of "order parameters", where accessible characteristics are purely global ones, or from an *internal* point of view, where accessible characteristics are limited to local ones. It can also be described from *a quasi-external* platform as an externally viewed set of internal relations. This latter picture corresponds to just about *every* system analysis which we carry out, but unfortunately in a system which exhibits *real* scale effects internal detail is *inaccessible* through the application of formal rationality, or at the very least only approximately (although conventional science commonly presupposes this not to be the case - and no, we have not forgotten quantum mechanics here!).

So, our most usual "quasi-external" view is self-contradictory in non-formally-rational systems! We cannot equate the properties of different scales of an even marginally non-formally-rational system, or even arbitrarily change viewpoint within one and the same scale level without addressing the associated information transformations. Working from the simplifying presupposition that nearby viewing platforms will most resemble the one we are currently standing on, we will try to approach this problem by distinguishing between *directly* and *indirectly* accessible inter-elemental system connections[1]. *Direct* relationships are established by inter-elemental negotiation of both rationality and context, directly and intimately between the elements concerned, as shown in Figure 1. *Indirect* relationships between elements are those which of necessity pass through other intermediate elements, which may then be free to impose their own modifications on forwarded information: choice of the viewing platform imposes an asymmetry on the resulting view. This problem makes an appearance even in very simple systems: it is the basis of the difficulty most usually referred to as the Newtonian three-body problem.

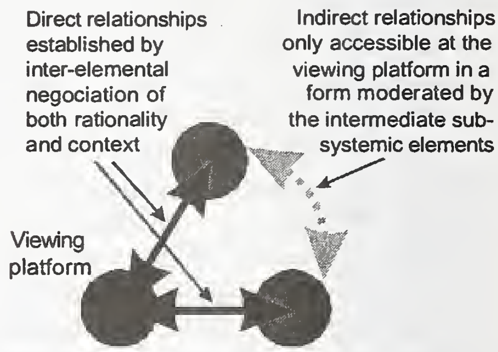


Figure 1. Direct and indirect relationships for a 3-body system with a chosen platform.

We can extend this distinction of direct and indirect linkages to larger ball-and-stick system models. Given 2 elements, we will have 1 direct link and no indirect ones; with 3 elements there will be 2 direct links and 1 indirect; with 4 elements, 3 direct, 3 indirect; with 5 elements, 4 direct, 6 indirect, and so on. As we move to larger randomly-connected systems the relationship between

[1] Clearly, the criticism we make here of "quasi-external" viewpoints can equally be applied to the argument we are ourselves presenting; but not, so far as we are aware, in a manner which leads ultimately to its destruction: we are trying to present a conceptual argument, and not a set of formally related parameters.

direct and indirect links takes on a clear form: the number of direct links goes up as the number of elements N; the number of indirect links goes up as the square of the number of elements $N^2/2$, as shown in Figure 2 (note that this effect is to some extent alleviated in scale-free networks [1], but that it never entirely disappears).

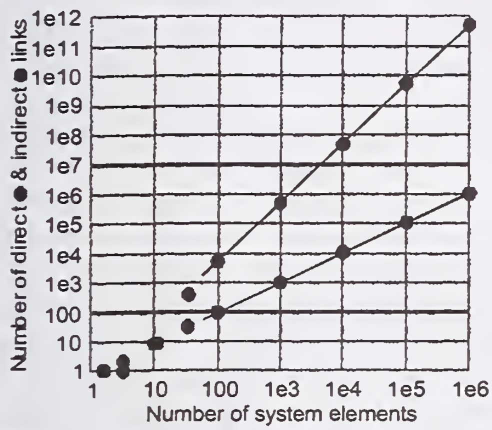


Figure 2. The growth of direct and indirect links in a large multi-element system.

The populations of direct and indirect character co-evolve at very different rates. For a system with one million direct links, there are a possible million-million indirect ones: for large systems indirect links are likely to dominate massively, depending on the complexity of the relationship between local and global structures. The character we can attribute to a complete system is ultimately *controlled* by this direct/indirect balance, as is the robustness of a network with respect to reductions in interconnection viability. The co-evolution of direct and indirect relations in large systems leads ultimately to two different independent systemic characters. One corresponds to the "normally scientific" view, which depends on formally-rational cross-scale information transport, the other corresponds to parts of the holistic system which are inaccessible to a "normally scientific" viewpoint, and which are associated with the distributed nature of indirect relations. Complete representation of systemic interactions with an environment requires the evaluation of *both* of these characters. If we simply describe a quasi-externally viewed system in terms of the reductively specified interactions we risk missing out *the majority* of the systemic character! (except if we are dealing with time-independent (clocked) artificial formal "machines" such as idealized digital computer systems). We believe that it is this bifurcation of systemic character into dual reductive and holistic parts, and the difference in rational accessibility between the two systemic characters, which has led to the conventional split between *body* and *mind*, where the body is automatically associated with direct "scientific" bio-systemic relations and the "mind" is naturally "difficult" to understand within the context of a "normally scientific" rationality which presupposes that all essential systemic aspects can be related to a single localized platform.

2. Hierarchical Stepping Stones

Large systems exist between two extremes: their unification as a single entity and the assembly of their smallest components. In any system, natural or artificial, where the spread in scale between these extremes is very large, intermediate self-supporting descriptive levels emerge (or are created) to facilitate transit across the entire scale-spread of the system (e.g. stairs, semiconductor inter-band traps, VLSI design, stars in the universe, ...). This aspect of large systems is so pervasive that we can formulate a universal model for the resulting hierarchical systems [2] (Figure 3), whose properties are very closely tied in with the arguments of Section 1 above. Here each level of the coupled "model" hierarchy represents one and the same entity, for example a tree. Successive hierarchical levels describe the entity with a progressively changing degree of detail, from the most elaborate to the most simple. For example: ..., a tree as atoms, a tree as molecules, a tree as cells, a tree as branches, a tree "as itself", ... These successively simpler representations of the entity "contain" progressively more and more sub-scalar detail which is hidden from locally-scaled ecosystemic interaction.

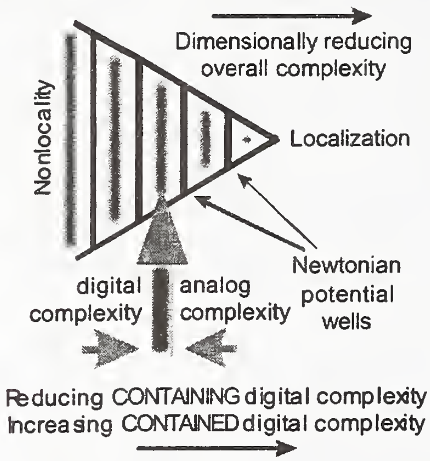


Figure 3. A generalized complementary hierarchical evolutionary system.

At a given level of the model hierarchy, the relevant representation provides a partial "en-closure" of sub-scalar detail, leading to simplification of the relationships between the entity and its locally-scaled environment as a trade-off against representational precision. Possibly the most important aspect of natural hierarchies is just this, that through the establishment of a series of related progressively abstracted models of low-level detail, high level "forms" are not constrained to operate within the complex temporal limitations of their low-level interactions. The apparently most simple model is the one which contains the most hidden sub-scalar detail. Within a computational paradigm this "hiding" of sub-scalar detail makes it possible for biological organisms to develop mechanisms for multi-temporally-scaled reactions to external stimuli, promoting survival in a complex hostile environment. Our own brains use a mechanism of this kind in the context of "fear-learning" [3].

The model levels appear as Newtonian potential wells in an otherwise non-Newtonian multi-dimensional and multi-scaled phase space between nonlocality (on the left) and localization (on the right) [4]. Moving between adjacent model levels, we encounter both kinds of complexity (digital and analog). Towards the left hand side of the assembly models are related to a global conservatism, and towards the right hand side to a local causality: the assembly forms a coupling structure between these two aspects of nature. Movement through different model levels towards the right corresponds to a reduction in the *containing* digital complexity of models and an increase in the *contained* digital complexity. It is worth adding that the Newtonian potential wells which correspond to the different model levels are regions of the universal phase space where global and local effects are self-consistent. This is a *fundamental* aspect of the stability and computability of nature. A major consequence is that, within these Newtonian regions, local causal interactions can proceed within limited temporal scales without fear of contravening a more global conservatism. However, the viability of such a structure as a general model of hierarchical systems depends on a fine balance between the isolation or "en-closure" of adjacent levels with respect to each other and the degree of inter-correlation which is necessary to support their stability and that of the hierarchy: a degree of inter-level correlatory information transport is *vital*: too much is fatal!

3. Ecosystemic Rationality and Emotion

The significance of the general hierarchical assembly we propose is that it represents the intermediate structure which we can observe, not only between the complementary pair of "extra-real" nonlocality and "extra-real" perfect localization, but between and internal to *any* and *all* high-level "extra-real" complements. We can find complementarities of this kind everywhere as soon as we start looking for them: quantum and classical descriptions; organisms and their ecosystems; ... There is, however, a further *complementary* twist to the story. The hierarchical assembly we propose dissociates into two distinct and complementary systems of *rationality*.

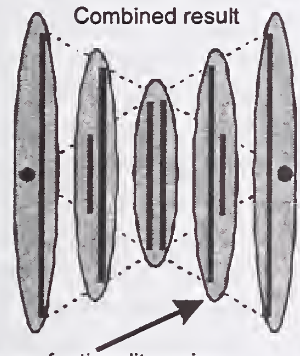


Figure 4. Interleaved "normal" and "complementary" rationalities in a generalized birational entity-ecosystem representation associated in local birational pairings.

One, of "normal" rationality, is associated with the Newtonian potential well model levels, and is reductive towards perfect localization. The other, of "complementary" rationality, is associated with the intermediate complex regions, and is reductive towards perfect delocalization [5]. The two systems are interleaved to give the complete structure which we showed earlier in Figure 3. It does not appear accidental that this binary complementary structure matches that of quantum-holographic vector-reconstruction information processing [6]. The result is a set of low-level local complements where a "normal" (local) level is always associated with a (local) ecosystemic "complement" level, as shown in Figure 4. The summation of both levels *at any scale* results in complete systemic description: the ecosystemic level provides a local but "normally inaccessible" store for all of the information which is eliminated through the "formally rational" compression to the current level through the stepping stones from the lowest level description (see the argument about a Boolean AND gate in Section 1). Interestingly, and probably unavoidably, this situation corresponds closely to the two sets of different information which are invoked during quantum error correction techniques.

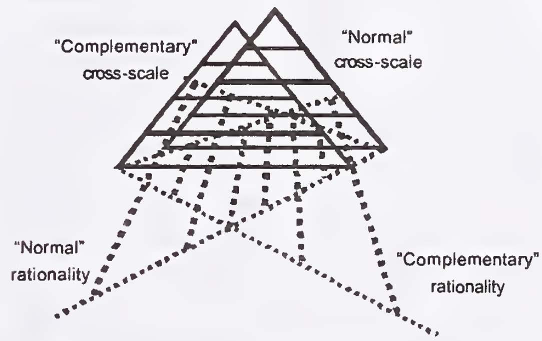


Figure 6. Interleaved "normal" and "complementary" multi-meta-scaled cross-scale rationality systems, based on their interleaved "normal" and "complementary" model assemblies.

We should remember that the Newtonian hierarchy must be globally stabilized by interactions right across the system between all scales. Noting that by "understanding" we usually imply "seeing the relationships between the level we are talking about and both higher and lower adjacent ones", we believe that inter-scalar interactions generate first a "hyper-scale" descriptive level, and then progressively a hyper-scalar hierarchy superimposed on the initial scalar one. A similar interaction for the "complex" hierarchy results finally in a *pair* of hyper-scale hierarchies, as shown in Figure 5.

It should be noted that at the highest hyper-scale levels, correlation between the developments of the two systems becomes increasingly less relevant, as the two structures progressively separate from each other in association with the degree of their individual abstraction. At the highest levels, each of the hyper-scale manifestations is clearly and distinctly independent, and identifiable with a different kind of rationality [7].

This corresponds to the conventional picture we hold of our own thought processes; on one side there is a "rationality" which corresponds to scientific logic, on the other a very "difficult-to-categorize" but *effective* "irrationality" we refer to in terms of "emotion". The two are complementary: failure of (rational) logic in pragmatic situations can be circumnavigated by recourse (irrational) emotion; failure of an emotional approach can be rectified by the application of logic. Neither one can successfully exist alone: reason needs emotion, emotion needs reason. Our civilization habitually focuses on only *one* of the pair: reason is everything; emotionally, even, we feel we "should" be logical!

4. Information Integration and Consciousness

As we all (*sic*) were taught in school, the mammal eye works to create an inverted image of the viewed scene at the retina (whose orientation is rectified by the brain). Not so. There is no integrative capability at the retina to perform this function. Any "image" is generated much later, in the various layers and centers of the brain: it only "exists" within the (abstract) unification of high-level consciousness, and *never* in any "real" sense describable by science. If you are viewing this text via a computer screen or through the printed word, the same constraint holds: it does not exist *at all* as a unified entity outside your brain or imagination, merely as a collection of informational elements devoid of any *implicit* organization, which was transmitted through the Internet by a means which has been formally (scientifically) structured through the application of our imagination to achieve our aim of reproducing arbitrary "patterns" across space and time. The same argument holds for the *entirety* of our environment: it is *all* beyond representation by (current) science. Not only does this description apply to "objects", it applies equally well to any and every subject of discussion.

Most particularly, in the current context of interest, we should not expect to find that a robot is capable of responding as a "black box" to external stimulus on the basis of an internally integrated "motive", except where that "motive" is completely relatable to its formally unified degenerate representation – namely the binary "it exists" or "it doesn't"! Such a quasi-hierarchical relationship (along with any "algorithmic" complexity it exhibits) is both nominally and functionally trivial when compared to the styles of *real* complexly-hierarchical operation which characterize living organisms. We should consequently beware of attributing anthropomorphic integrative unification to the internal workings of a "black box" robot unless it is *entirely* predictable (a character which corresponds *exactly* to the quasi-hierarchical condition referred to above), in which case any resemblance of its actions to those of a human is far from likely, to say the least! So, can we describe and develop robots "in our own image" by the application of scientific techniques, or not? Or does the problem which must be addressed reside elsewhere?

Descriptions of the natural world and the placing of robots within it which derive from Evolutionary Natural Semiotics (ENS) by way of *signs* are untouched by this dilemma. In the context of ENS, any formalized representation is derived pragmatically (but less-than-algorithmically) from its own scale-local grounding, and the various scale-localizations are coupled through and within the context of a global-to-and-from-local correlation which mediates between the scale-local groundings of a global grounding which it also creates (!). “Reality” (in a scientific reductionist sense) then refers to nothing more that the lowest level of description which we can be bothered to deal with, whether that be the atomic level, super-strings, membranes, people, trees or psychological states.

The descriptions which we habitually employ for “systems” which are internally structured in a network-like manner are suitable if, again, the network structure is amenable to complete (formal) integration *reductio ad adsurdum*, but for a “system” which exhibits “useful” complexity, they are worthlessly simple or simplified. Within ENS such representations (where we view the “system” as a whole and *simultaneously* its network-like internal structure) have the character of (dubious) quasi-external representations, whose (cautious) applicability depends primarily on their degree of representational equilibrium. Much effort is currently being expended in developing “internalist” models of operational situations, rather than the “externalist” ones said to be characteristic of scientific endeavor. It is difficult to imagine, however, how a uniquely internalist representation of a “conscious” or aware state can or could be useful: its existence would imply not only the usually-quoted criterion of lack of knowledge of the causes of received stimuli, but also the complete absence of any attempt to investigate or imagine the origins of those stimuli. To investigate in such a way requires the construction of an (imagined) externalist model of the situation: to not do so seems to imply lifelessness! Consequently, it makes more sense to describe living interactions as a negotiation between internalist and externalist representation, through a process which mirrors the internal-external negotiations which lie at the roots of human consciousness and moderated autonomy [8].

Human consciousness is “singular”, in that it only exists as an individual unified “entity”. It is within the “sufficient interpretation” and correlation of a multiplicity of informational details that *this* text *becomes* (nothing more… just “*becomes*” itself) within our consciousness. Its existence emerges from the process of integrative interpretation (or interpretive integration, if you prefer). This process, of the emergence of the informal from the formal (simplistically describable as emergence of the *analog* from the *digital*), is the very nature of living entities. It appears most obviously, but not uniquely, in the generation of analog protein folding from the digital code of DNA. Science does not merely *omit* this emergence from its confines; it *expels* it, as being too difficult to deal with. A lifelike nature is by definition external to a scientific development!

Cooperative Intelligent Systems

So, how are we to develop “lifelike” cooperative intelligent systems? Ultimately, not through uniquely digital computation, although this can provide effective interfacing between a central information processor and the outside world. This is itself the manner in which our own brains operate: a central *really* parallel processing style, whose operation is most closely related to the superposition-and-selection mechanisms of quantum mechanical interaction [9, 10], and integration and differentiation of the results of this processing to serve localized output and input nodes. Currently this style of integration and differentiation is far beyond our constructional capabilities, and while a prime target must be to investigate and develop lifelike information integration, we can nevertheless achieve useful preliminary results if we couple our targets to the means which are available, so long as we do not fool ourselves into thinking that this will be sufficient.

Two routes (at least) present themselves for simulation of a desirable computational structure. One depends on the axonite mesh proposition of Karl Pribram [10], in which the outputs from a large number of sender-neurones are distributed in parallel as a quasi-wave to a large number of receptor-neurones, simulating the nonlocal distribution of solutions which characterizes quantum superposition, and storing the associated information in a distributed manner as a "collapse" of the wave at the receptor dendrites. This strong contender matches well with experiments carried out to define the neural location of consciousness [11]. The other route depends on the mathematical distribution of information across a large parallel processing network by the recursive integration of Dempster-Shafer probability into diffuse rationality [12]. It remains, however, difficult to see how either of these routes can provide a sufficiently "intelligent" information integration to generate any "real" consciousness in an artificial structure, and long-term hopes most probably rest with currently advancing projects which aim to introduce less-than-formal computation into the hardware elements of computer processing, rather than with the simulation of parallel processing via digital software.

References

[1] Barabási, A-L. "Scale-free Networks." 4th International Conference on Complex Systems, Nashua, USA, June 10-14, 2002. Forthcoming publication.

[2] Cottam, R., Ranson, W. and Vounckx, R. "A biologically consistent hierarchical framework for self-referencing survivalist computation." In: *Computing Anticipatory Systems: Casys'99 3rd Internat. Conf.: Liège, Belgium,* Ed. Dubois D. M., AIP Conference Proceedings, 517, pp. 252-262, 2000.

[3] LeDoux, J. E. "Brain mechanisms of emotion and emotional learning." Current Opinion Neurobiol., 2, 191-197, 1992.

[4] Cottam, R., Ranson, W. and Vounckx, R. "Localization and nonlocality in computation." In: *Information Processing in Cells and Tissues*, Eds. Holcombe, M. & Paton, R., New York: Plenum Press, pp. 197-202, 1997.

[5] Cottam, R., Ranson, W. and Vounckx, R. "Life as its own tool for survival." In: *Proc. 43rd Meeting of the International Society for the System Sciences*, Eds. Allen, J. K., Hall, L. W. & Wilby, J., Pacific Grove, USA: ISSS, #99-168, pp. 1-12, 1999.

[6] Schempp, W. "The DNA replication of molecular genetics: a control paradigm of the quantum holographic information channel." 4th International Conference on Information Processing in Cells and Tissues, Leuven, Belgium, August 13-17, 2001. Forthcoming publication.

[7] Cottam, R., Ranson, W. and Vounckx, R. "A diffuse biosemiotic model for cell-to-tissue computational closure." *BioSystems*, 55, 159-171, 2000.

[8] Cottam, R., Ranson, W. & Vounckx, R. "Consciousness: the precursor to life?" In: *The Third German Workshop on Artificial Life: Abstracting and Synthesizing the Principles of Living Systems*, Thun: Verlag Harri Deutsch 1998, pp. 239-248, 1998.

[9] Cottam, R., Ranson, W. and Vounckx, R. "Emergence: half a quantum jump?" *Acta Polytechnica Scandinavica*, Ma 91, 12-19, 1998.

[10] Pribram, K. H. "Proposal for a quantum-mechanical basis for selective learning." 4th International Conference on Emergence, Odense, Denmark, July 31 – August 4, 2000. Forthcoming publication.

[11] Tononi, G. and Edelman, G. M. "Consciousness and complexity." *Science*, 282, 5395, 1846-1851, 1998.

[12] Cottam, R., Ranson, W. and Vounckx, R. "Diffuse rationality in complex systems." *Interjournal of Complex Systems*, Art. 235, 1998.

Integrating Effective Planning Horizons into an Intelligent Systems Architecture

J. P. Gunderson and L. F. Gunderson
Gunderson and Gunderson, Inc.

ABSTRACT

One metric of the intelligence of a system is its ability to perform tasks in the face of dynamic changes to the environment. This requires that an autonomous system be capable of responding appropriately to such changes. One such response is to effectively adapt the allocation of resources from planning to execution.

By adapting the resource allocation between cognition and execution, an intelligent system can produce shorter plans more frequently in environments with high levels of uncertainty, while producing longer, more complex plans when the environment offers the opportunity to successfully execute complex plans.

The effective planning horizon is developed from an analysis of mathematical models of classic autonomous system and from current research in cognitive science. Experimental results are presented showing the performance gain from an effective planning horizon based system.

From this simplified feedback control model, the Effective Planning Horizon concept is extended to a more realistic intelligent system architecture, and the concepts of bounded rationality, intelligent heuristics and the judgment analysis "lens model" are shown to be analogs of the effective planning horizon.

Keywords: *Effective Planning Horizon, Interleaved Plan/Execution, Probabilistic Planning, Intelligent System.*

1 Introduction

The world is not a perfect place, and if intelligent systems are to function effectively they must be capable of handling both the uncertainties of the world outside themselves and those created by their own limitations. One working definition of intelligence is the ability to adapt to unexpected failure. Using this definition, intelligence is defined by the ability of the system to achieve goals in the face of failure. In order to do this, the system must have some mechanism to detect that its current method for satisfying its goals is insufficient, and to select and apply new methods.

In this paper we present the concept of an effective planning horizon as a requirement for intelligence. We start by discussing the reasons that an effective planning horizon is necessary. We place the model of an effective planning horizon into a model of an intelligent system. Then we develop a mathematical definition of the effective planning horizon. This definition is tested in the domain of a simulated Autonomous Underwater Vehicle. In the final sections of the paper we present arguments to show that the effective planning horizon concept can easily be transformed into its analogs in sensing, acting, and goal selection.

2 Why an Effective Planning Horizon

Why can't an intelligent system just make one plan? The answer to this lies in the balance between the complexity of all environments and the limited computational resources that any intelligent system can apply to the problem.

It is obvious that all natural environments and all but the simplest of synthetic environments are stochastic. Even in a simple deterministic environment, the observations made by an entity have a probability distribution [1]. As demonstrated below, if the environment is not deterministic, then the longer the plan needed to meet the goal, the less likely it is to succeed. However, even in an interleaved planning-execution system, too short a plan length will often lead to sub-optimal plans.

Alternatively a system could generate all conceivable plans in advance and maintain those in a plan library [2]. But, in any realistic domain, it is not possible for an intelligent system to completely explore all the possible ramifications of a plan of action in a realistic time [3]. In addition, to time constraints, there are memory constraints. In fact for one nominally intelligent system, only 7±2 objects can be held in working memory at any one time [4]. This seriously limits the number of plans that can be considered. This has led to the concepts of "bounded rationality" in both biologic [5] and machine based intelligence [6].

Thus, it appears that all intelligent systems (whether biologic or electronic) must make trade-offs between cognition and action. There are many mechanisms for placing bounds on cognition, and in this paper we focus on the concept of a planning horizon.

3 A Model of an Intelligent System

The model we use is based on the elementary loop of functioning (E.L.F.) model of Meystel and Albus [6]. This model assigns responsibility for four critical tasks to independent processes in an iterative - *sense*, *model*, *plan*, *act* loop. We have extended this model by adding a new process, *validate*. This new process has the ability to compare the observed result of the previous cycle with the expected result. It then updates the other processes with the observed success or failure of the cycle. This is show graphically in Figure 1.

This model allows for the representation of intelligence as a semi-lattice. In this representation, the intelligence of a system is defined by the ability of the system to perform each of these processes. We utilize goal satisfaction rates as a primary metric of intelligence. The intelligence of the system is a 5-tuple of the intelligence of each of the processes. The intelligence of each of the processes is incommensurate with the intelligence of the other processes, thus establishing the partially ordered set needed for the lattice.

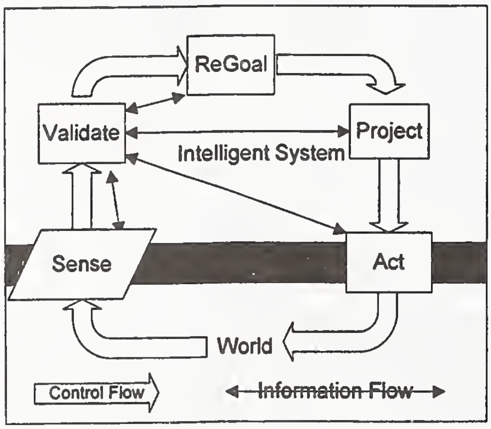


Figure 1 Extended model of Intelligent Systems

4 Intelligent Systems

Since the earliest work in machine intelligence, a concern has been the computational burden required for 'intelligent' behavior. With the representation of planning as search [8] came the exponential growth of the number of *world states* explored with the increase in search depth. The number of world states explored is a measure of the computational complexity of the planning process. It has been long accepted that as uncertainty and dynamism increase in an environment, the need for more reactive systems also becomes greater [9]. In recent work, quantitative relationships have been suggested for the impact of three types of domain uncertainty on the ability of an autonomous system to achieve goals.

This work classifies domain uncertainty into three categories:

Sensor uncertainty;

Effector uncertainty; and,

Uncertainty caused by exogenous events.

These three types of uncertainty have been used in a simulation system that measures the goal satisfaction of a simulated maintenance robot under widely varying levels of each of these types of uncertainty [10].

These results suggest that autonomous systems are very sensitive to even low levels of uncertainty in the environment, with overall goal satisfaction dropping to 75% with the introduction of 10% sensor errors. In addition, this research discovered that the ability to retry on failure was very effective at allowing the system to maintain goal satisfaction in the face of all types of uncertainty. The simulated robot was more robust (i.e., able to satisfy goals as the environment changed) when it used different deliberation and execution mixes in response to different levels of uncertainty in the environment.

5 Defining the Effective Planning Horizon

Beginning with early two-player game models, it has been common to place a limit on the search depth used to explore options. This was often implemented as an arbitrary limit on the ply-depth of an alpha-beta search tree.

In two player games of perfect information, it is implicitly assumed that if the computational resources were sufficient, the ply-depth could be increased until a perfect game analysis was performed. However, when an intelligent system is playing against the real world, conditions change. Since there is an inherent variability in the world, perfect information is impossible. Under these conditions, there appears to be a hard limit on the planning horizon – a line beyond which planning is useless, since it is impossible to know the state of the world in which the planning is taking place.

We approach the idea of an effective planning horizon from the perspective of the expected value of additional planning. What is the value associated with extending my plan, and what is the likelihood that the world will be in the state I planned for?

$$E(plan) = \Pr(plan) * Value(Plan) \qquad (1)$$

While solving the exact value of equation 1 may be impossible, it is possible to make some reasonable assumptions about the terms, and from these assumptions draw some conclusions about the characteristics of the expected value of planning in dynamic and uncertain domains.

We make three assumptions about the actions in a plan:

1. The action achieves some state that is needed to achieve a goal (or goals);

2. Each action has some finite probability of failure; and,
3. All the actions in the plan must succeed for the plan to succeed.

Under these assumptions it is clear that a first approximation of the probability of the plan succeeding is:

$$\Pr(plan) = \prod_{steps} \Pr(Step) \qquad (2)$$

This can be further simplified by assuming that all actions have some common probability of success, reducing Equation 2 to:

$$\Pr(plan) = \Pr(step)^{length} \qquad (3)$$

Approximating the value associated with a plan of given length is more problematic, however for any given domain and set of possible goals to achieve some useful assumptions can be made.

First, let us envision the complete range of problems that our intelligent system might be required to solve. This range could be defined as the cross product of all possible initial conditions with all possible final states. Then let us imagine the plans that might be utilized to transform the world as we find it into the world as we desire it. Clearly, very few of the (Initial, Final) pairs can be transformed by plans utilizing a single action. A few more, perhaps, can be achieved with two-action plans, more yet by three action plans, and so on. If we further take the view that we wish the simplest, or most likely, plan to satisfy our needs, then there will be few pairs that require 1000 action plans, and fewer still that require 2000 action plans. If we model the value of a plan of size N as the number of (Initial, Final) pairs that can be satisfied by a plan of length N or less, we get a sigmoidal, or logistic curve. This curve can be closely approximated with an equation like (3) below:

$$Value(plan) = 1 - \frac{a}{1 + 2^{length}} \qquad (4)$$

Finally, combining the two terms into the expected value of a plan of length N, for a given domain, and task assignment, we find that the resulting curve has the characteristic shape shown in Figure 2.

This finding is consistent with the analysis that, there exists some point beyond which the costs associated with planning exceed the benefits, which causes the expected value to decrease. Under the assumptions made above, we can conclude that there is a range of plans, which provide the optimal benefit to the deliberative system. This range is defined as the effective planning horizon (EPH). If the planning depth is less than the EPH, the probability that effective solutions will be produced is too low. If the planning depth exceeds the EPH, the probability that significant amounts of computational resources will be expended planning for situations that never occur is high. If those resources had been applied more effectively, the rate of goal satisfaction could increase. However, while this analysis suggests that it is beneficial to adjust the planning horizon to the domain and goals of the intelligent system, it does not provide any mechanism to achieve this adjustment.

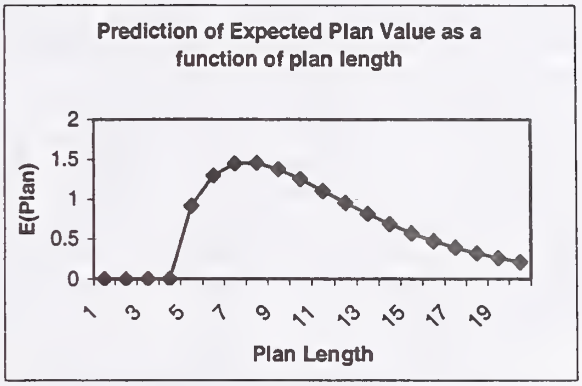


Figure 2 - Expected Value of Planning

5.1 Methods of Adjusting the Planning Horizon

In this paper we investigate a method of selecting an effective planning horizon, adapted directly from the notion of placing a limit on the search depth. The planning system used is an interleaved, probability-aware, forward chaining planning and execution environment. It is a general-purpose planner in that all domain specific information is encoded as part of an input file. Included in this domain specific information are naive probabilities of success for the available actions. The planning system uses this information to produce the plan with the highest observed probability of success.

These experiments were run using an autonomous underwater vehicle domain. In this domain the planner is embedded in the autonomous underwater vehicle (AUV). The types of tasks assigned to the AUV include autonomous navigation, the deployment of monitoring devices within enemy controlled territory, and avoiding or escaping detection by enemy Anti-Submarine Warfare (ASW) surface vessels.

A typical task assignment would be to:

1. Begin from Home base, at the surface, carrying a deployable monitor;

2. End at Home base, at the surface, with the monitor deployed at a location just outside the mouth of the enemy harbor; and,
3. By the way, don't get detected by any enemy ships while you do it.

If the system has perfect knowledge, and nothing changes, this task is a straight forward planning exercise. However, in this domain, the enemy ship moves while the AUV is executing its mission and the AUV has limited sensor range. This forces the planner to develop a plan, begin to execute it, and respond to failures. These failures could include finding the ASW ship in its path or being detected during a move. More complex failures might involve having to switch goals to escape detection and then re-target the goal of dropping the monitor, once it has lost its opponent.

The planning and execution system is interleaved, so during the execution of actions, it receives feedback about the actual state of the world (at least those parts which it can sense) and it compares the actual state with its expected state. As long as the expected state of the world and the actual state agree, the system continues to execute the planned actions. If the states do not match, several options are available:

1. Continue with the current plan,
2. Develop a new plan to meet the current goals,
3. Re-evaluate the goals, and develop a plan to meet the new goals, or
4. Do nothing, and hope that the world changes.

After this evaluation, a new action is issued, and the process begins again. Since the planner is deeply embedded in the execution loop, strict limits on computation must be imposed to assure responsive behavior of the system.

5.2 Search Depth Control

Search depth control has been used to limit computation for as long as planning as search has been used. In general there are two forms of this control: limiting the exploration depth directly – such as ply-depth limits, or plan length limits- and world set limits. In the latter case, the planning system monitors the total number of individual worlds explored, and at some pre-determined limit, stops exploration. This has benefits when coupled with search control rules which allow the planner to 'focus attention' on areas of the plan which might be more fruitful. The experimental planning system used here uses this world count form of search depth control.

To explore the impact of limiting search depth on plan success rates the senario is analyzed under a range of allowable search depths.

6 Experimental Results

The experiments were designed to answer the question "For a given domain, and a given task mix, at what point does it stop paying to plan?" Using the analysis in Section 5, we measure the marginal value of planning by measuring the success rate of the plans produced, and the computational complexity of producing the plan. All experiments are run on the same interleaved planning and execution system, coupled with an external simulator of the domain.

6.1 Experiment Setup

The domain used is that of the AUV, described above. The AUV can navigate in a world of 18 possible locations, at up to three depths for each location. Any motion has a risk of causing the AUV to be detected, however the deeper the AUV the lower the risk, In addition to movement actions, the AUV can change depth, deploy a monitor, and cause a tracking ASW vessel to lose track by going deep and drifting. The AUV carries one monitor, and the task requires the AUV to deploy that monitor at a specific location and return to base undetected.

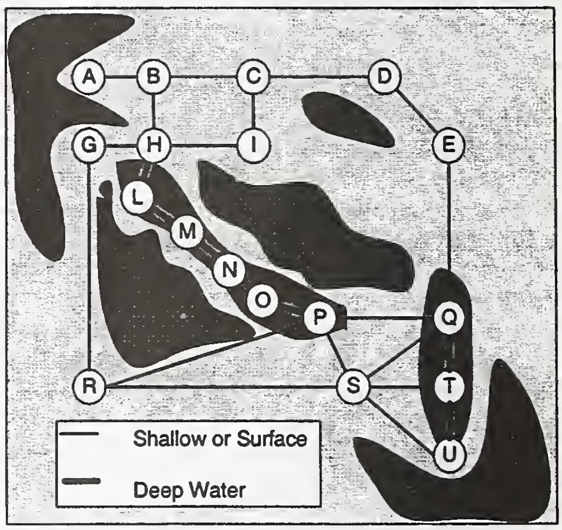


Figure 3 - Autonomous Underwater Vehicle domain

The task assigned to the AUV in these experiments is to travel from home base (Location A) to Location P, release a monitor, and return to Location L for pickup (sub-lp10.prob). It has the additional requirement of remaining undetected during the mission, and should it be detected it must cause the enemy vessel to lose its lock before being picked up (Figure 3). The AUV has six general types of operators to apply:

Table 1 Relative probability of success for the AUV domain. Failure can mean either the AUV is detected, the operator fails, or both.

Operator Type	*A priori* Probability of Success

Travel Surface	0.45
Travel Shallow	0.90
Travel Deep	0.95
Change Depth	0.99
Release Monitor	0.99
Drift (can cause opponent to lose detection)	0.25

For the assigned task, the naïve probability of success is approximately 0.46, if the plan with highest probability is selected, or 0.00015 if the plan with the minimum number of operators is selected. One additional complicating factor is the behavior of the ASW vessel. This unit moves around the map in a random walk, and if it occupies a location it blocks the AUV from traveling to that location. The location of the ASW is initially unknown by the AUV, however, the AUV can sense the presence of the enemy vessel if it enters or leaves an adjacent location.

A single simulation consists of assigning a task, executing the mission, and recording the number of individual world states explored by the planner during the execution of the mission and the success or failure of the mission.

Two baseline simulations are initially run. In the first the planning horizon required for success by a non-interleaved planning system is determined. Since a non-interleaved system must be able to produce a single plan to meet all the mission goals, the required search depth is much greater than that required by an interleaved planner. In this experiment the planning horizon is increased until the planning system reaches the point where it can produce a complete, successful plan. This established the minimum planning horizon for a non-interleaved planner. Since the planner is only allowed to execute a single cycle, the success rates of this plan cannot be effectively compared with the success rates of the interleaved simulations.

The second baseline addresses this problem by using the planning horizon established by the first baseline. However, the planning/execution system is run in interleaved mode, allowing the planner to correct failures. This second baseline establishes the plan success rate and the minimum number of world states required for an interleaved planner operating without a search depth limit.

The experimental simulations use an interleaved planning system, with a range of planning horizons. In all experiments a minimum of 100 independent simulations are run at each planning horizon. In interleave mode, the system is allowed to compete up to five planning/execution cycles. The average number of world states examined during the each simulation and the average success rate of the mission are recorded.

6.2 Data Analysis

The data analysis is straightforward. We calculate the value of the planning horizon (PH) as:

$$Value(PH) = \frac{SuccessRate}{MegaWorldsExplored} \quad (5)$$

The value is the success rate over the work expended (million worlds explored to achieve the success rate).

Baseline 1 established that it is necessary to examine approximately 13,000 world states, and the success probability is 0.14 for the non-interleaved planning system. Due to the interaction with the ASW vessel, this is significantly below the naïve probability (0.46). In effect, most of the computational resources used by the intelligent system are expended planning for situations that simply do not occur during the execution. For Baseline 1 the Value(PH) = 10.76.

Using the planning horizon established by Baseline 1, Baseline 2 achieves a success ratio of 0.74. However it explores 66,000 world states, for a Value(PH) = 11.21. Interestingly, even though the success ratio increased, the number of world states explored did so proportionally. This data is presented in Table 2.

Table 2 Baseline performance of complete planning system. In Baseline1 a single iteration of the planning system was allowed, in Baseline2 five iterations were completed.

Test	Worlds	Success Rate	Value(PH)
Baseline1	13,000	0.14	10.76
Baseline2	66,000	0.74	11.21

The experimental simulation was run with the range of planning horizons shown in Table 3. Note that the peak value of value(PH), 68.7 is approximately 6 times the value achieved by Baseline 2. This shows that at the effective planning horizon, the planning system is achieving equivalent success rates with significantly less computational cost.

Table 3 Table of the computational resources required for varying planning horizons, the achieved success rates, and the Value of the invested computational resources.

Planning Horizon	Worlds	Success Rate	Value(PH)
100	9,605	0.02	2.08
150	14,200	0.04	2.81
200	18,854	0.06	3.18
250	12,813	0.88	68.7

300	14,586	0.88	60.3
350	16,717	0.91	54.4
400	19,358	0.89	46.0
450	20,026	0.94	46.9
500	22,315	0.90	40.3
550	22,726	0.93	40.9
600	27,615	0.86	31.1

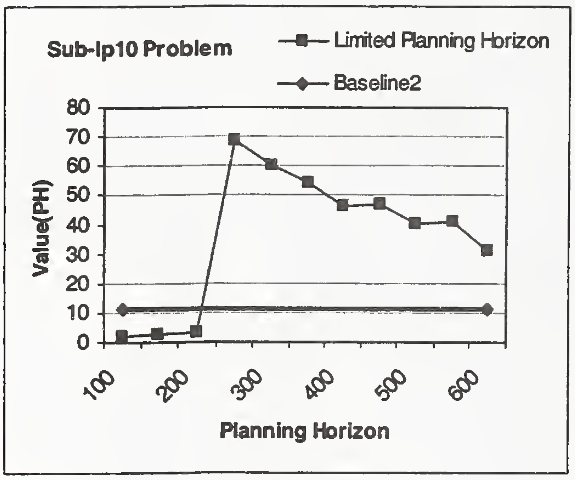


Figure 4 - Plotted Value of Effective Planning Horizons from the AUV problem

In Figure 4, the value of the planning horizon, as defined in Equation 5 is plotted against the search limit imposed on the planner. The first three points are characteristic of planning horizons that are too short to allow the planning system to find successful plans. The range from 250 – 350 represents the range of the EPH where successful plans are being found with minimum computational investment. The range above 350 is characteristic of wasted resources, planning for situations that never occur.

The effective planning horizon is the global maximum of this curve. It should be noted, that this figure closely resembles the analytic solution shown in Figure 2. The "Baseline2" line is the value achieved by Baseline 2.

It is clear that limiting the planning horizon in interleaved intelligent systems allows the system to achieve success rates which are comparable to complete planning systems, at a fraction of the invested resources. While the current data results from adjusting the search limit utilized by the planning system, several other mechanisms exist to adjust the planning horizon, including waypoint based planning [11], and the use of assumptive systems [12], which was successfully applied during the 1994 AAAI Robot Contest.

7 Extension to a Complete Intelligent System Model

The preceding discussion describes the benefits of an effective horizon in planning. However, this argument can be extended to all of the other processes.

Sensing
Validation
Re-Goaling
Acting

In the next section we sketch some ways in which effective horizons might be achieved for these processes. Much of this discussion will be derived from work done on biologic systems.

7.1 Effective Sensing Horizons

One model of the sensing process can be taken from the work of Egon Brunswik [1]. In this lens model, shown graphically in Figure 5, perception can be modeled as a linear weighted sum:

$$y_s = \sum_{i=1}^{n} w_i x_i \qquad (6)$$

Where y_s = the judgment of the condition of target s
y_e = the actual environmental condition of the target
n = the total number of cues available to the judgment maker
x_i = value of cue i, where i goes from 1 to n
w_i = the weighting of cue i

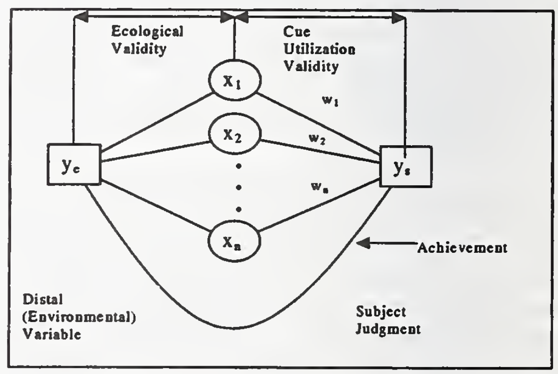


Figure 5 - The Lens Model of Egon Brunswik

For this process the effective sensing horizon will be a balance between using so few cues that the perception in invalid and using so many cues that the cognitive system is overloaded.

7.2 Effective Validating Horizons

Validation of the successful completion of a cycle is critical if the intelligent system is to adapt to unexpected outcomes. However, given limited sensing ranges, and long distance effects, it can be resource intensive to do complete validation. In some cases, the system may have to develop and achieve complete new sub-goals to accomplish this phase. Yet, the costs of proceeding without validation can be extreme. Thus any intelligent system must strike a balance between the costs and benefits of validating the world state.

In addition, when things go wrong, an intelligent system must assess the probable causes of the failure, and make improvements or the same ineffective actions will occur again and again. If the mission failed because a sensor was incorrectly interpreted causing a specific action to fail, it is not intelligent to expend resources improving the action. However, correctly attributing the source of a failure is extremely resource intensive, and limits must be placed on its application.

7.3 Effective Goal Re-prioritization Horizons

Intelligent systems do not pursue the same goals at all times. Consider the mother alligator, who under normal circumstances has a hair trigger bite reflex. Yet, this same alligator will carefully carry its young about cradled in those same jaws. Clearly, this intelligent system is re-prioritizing its goals in response to changes in its environment.

However, this ability to re-goal comes at a cost. Re-goaling requires the system to use cognitive resources that could be applied in other ways, and errors in goal prioritization can reduce the success probabilities of the system. This is the same dynamic tension that exists in the selection of an EPH.

7.4 Effective Action Horizons

Just as all biologic intelligent systems have limits on cognition, there are limits on both the number and range of actions the system can undertake to achieve a goal, and the quality of those actions. In biologic systems "Use it or lose it" applies, yet limited time is available for practice. With mechanical systems one tends to think of the range of actions that are available as fixed at the time of construction, and the quality (probability of success) as constant. However, as bearings wear out, and rubber gripping fingers age, the ability of the system to meet its goals degrades, until new resources are applied. While self-repair is beyond the current capabilities of most machine-based intelligent systems, they can possess the ability to update the reliability of actions to reflect changes in the system itself.

8 Conclusions

The focus of this paper is on utilizing limited computational resources to improve the effectiveness of intelligent systems. Drawing on research from existing biological intelligent systems and current research into machine-based intelligent systems, an analytic definition of the Effective Planning Horizon was developed.

Success probability in stochastic domains was presented as the key metric for evaluating intelligent systems, and several secondary measures including the expected value of planning, and plan value as a function of computational load were introduced as supporting concepts for the Effective Planning Horizon.

The EPH is a measure of the dynamic and variable nature of the environment, and the goals and limitations of the intelligent system. From these characteristics it is possible to establish a horizon beyond which additional planning is ineffective.

A simulated domain was presented which is representative of the types of tasks we expect deployed intelligent systems to undertake, and using a planning and execution system designed for these demanding domains, experimental data was collected. The data collected demonstrates that it is possible to achieve the same levels of success as the computationally expensive complete exploration of a plan space, at significantly lower cost. This lowered cost translates into lowered demands on the system, or increased speed of execution by the intelligent system, which can be crucial design requirements of future embedded intelligent systems.

Finally, the same principles which led to the formalization of the Effective Planning Horizon were applied to the other processes in feedback control theory based intelligent systems, suggesting other mechanisms that can be used to improve goal satisfaction by intelligent systems.

9 References

[1] Brunswik, E. *Perception and the Representative Design of Psychological Experiments*, University of California Press, Berkeley, CA, 1956.

[2] Schopper, M. J. "Universal Plans for Robots in Unpredictable Environments," Proceedings of the 10th International Joint Conference on Artificial Intelligence, 1987, pp. 1039 – 1046.

[3] Ginsberg, M. L. "Universal Planning: An (Almost) Universally Bad Idea," Artificial Intelligence, vol. 10, num. 4, pp. 40-44, 1989.

[4] Miller, G. "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *The Psychological Review*, vol. 63, pp. 81-97, 1956.

[5] Gigerenzer, G. and Todd, P. M. *Simple Heuristics That Make Us Smart,"* Oxford: Oxford University Press, 1999, ch. 1, pp. 3-34.

[6] Bratman, M. E. Israel, D. J. and Pollack, M. E. "Plans and Resource-Bounded Practical

Reasoning," *Computational Intelligence*, vol. 4, num. 4, pp. 349-355, 1988

[7] Meystel, A.M. and Albus, J.S. *Intelligent Systems Architecture, Design, and Control.* N.Y.: Wiley Interscience, 2002, ch. 2, pp 90-92

[8] Newell, A. and Simon, H. A., *Computers and Thought*, Fiegenbaum, E. A. and Feldman, J., eds., N.Y.:McGraw-Hill, 1963,

[9] Brooks, R. A. "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. RA-2(1), pp. 14-23, 1986

[10] Gunderson, J. P. and Martin, W. N. 'The effects of Uncertainty on Plan Success in a Simulated Maintenance Robot Domain," *Journal of Theoretic and Experimental Artificial Intelligence*, vol. 12, pp153-164, 2000

[11] Gunderson, J. P. and Martin, W. N. "Applications of Data-Mining to Sub-Plan Selection in Autonomous Planning Systems," in Proceedings of the IEEE Systems, Man and Cybernetics Conference, 2001, pp 1459-1464.

[12] Nourbakhsh, I. R., *Interleaving Planning and Execution for Autonomous Robots*, Boston: Kluwer Academic Publishers, 1997, ch. 7, pp 99-107.

MOBILE ROBOT POSE TRACKING FOR PERFORMANCE ANALYSIS

Alan M. Lytle*, Kamel S. Saidi, William C. Stone

Materials and Construction Research Division
Building and Fire Research Laboratory

Michael Shneier

Intelligent Systems Division
Manufacturing Engineering Laboratory

National Institute of Standards and Technology
**Building 226, Room B144, MS-8611, Gaithersburg, Maryland 20899*
alan.lytle@nist.gov, ph. 301-975-6048

ABSTRACT: The NIST Construction Metrology and Automation Group, in cooperation with the NIST Intelligent Systems Division, is researching robotic structural steel placement as part of a project to develop an Automated Steel Construction Testbed. The initial phase of this project centers on tracking a six degree-of-freedom robotic crane with a laser-based site measurement system to provide position feedback for autonomous steel assembly. Follow-on efforts will use a high-resolution LADAR scanner co-registered with the site measurement system to provide world model data. The combination of these two advanced metrology systems provides an opportunity for testing performance characteristics of mobile intelligent systems.

KEYWORDS: intelligent control, intelligent systems, performance metrics, 3-D coordinate measurement systems

1.0 INTRODUCTION

The NIST Construction Metrology and Automation Group (CMAG) is developing a robotic structural steel placement system for the testing and validation of advanced tools, methodologies, and standards for automated steel construction. This system, the first phase of the NIST Automated Steel Construction Testbed (ASCT), will demonstrate autonomous "pick and place" assembly of structural steel components using a six degree-of-freedom (DOF) robotic crane and an external pose estimator [1].

The base platform is the NIST RoboCrane, which is an inverted Stewart platform parallel link manipulator. The pose (position and orientation) estimator, a laser-based site measurement system (SMS), provides absolute cartesian position feedback to RoboCrane's Real-Time Control System (RCS) for trajectory planning and dynamic control. A world map of the robot work volume including target components and obstacles is created prior to operations using the SMS.

Future work on the ASCT will include incorporating a high-resolution LADAR (laser detection and ranging) system to create and update the world map, thus eliminating the requirement for a human operator to digitize the scene with the SMS prior to operations. The LADAR scans will be meshed and then registered to the SMS.

The use of an independent, external measurement system to track a mobile robot within an environment mapped and registered to the tracking system provides interesting opportunities for conducting mobile robot performance analysis. In a future experiment, the NIST Intelligent Systems Division (ISD), in cooperation with CMAG, will use this combination of metrology instruments to test the navigation and sensing systems on board the NIST robotic HMMWV test vehicle as part of the U.S. Army's Experimental Unmanned Ground Vehicle System (DEMO III) program.

This paper discusses the development of the robotic structural steel placement system and a proposed test methodology for the DEMO III sensor package performance analysis.

2.0 ASCT

2.1 Operational Concept

Four laser transmitters are positioned on the site perimeter to illuminate the work volume of RoboCrane with reference beams (Figure 1). A digital model of the construction plane including any obstacles is created using the SMS digitizing wand.

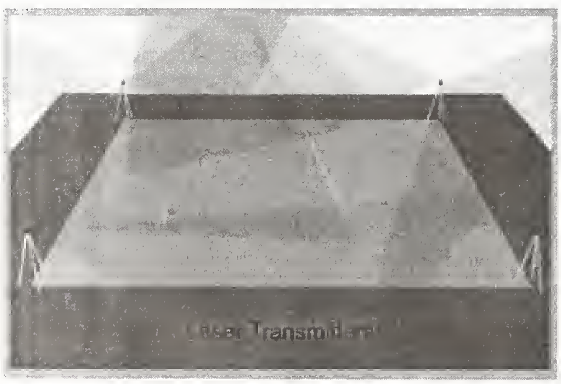


Figure 1: Illuminating the Work Site with the SMS.

This world model is then updated with the positions of the as-built structure and the target beam through a process of automatic part identification (barcode), part model database access, and part fiducial point measurement using the mobile digitizing wand (Figure 2).

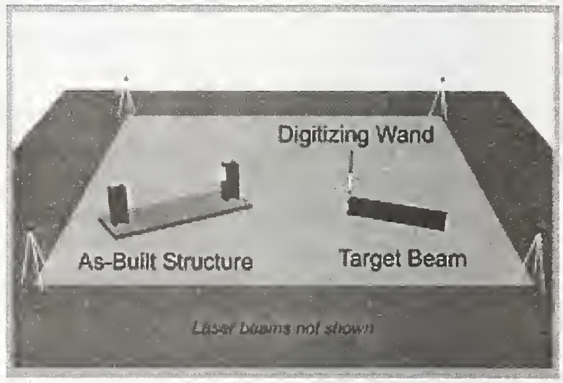


Figure 2: Measuring component locations with the SMS digitizing wand.

The current pose of RoboCrane is measured from onboard SMS sensors, and the path planner calculates the required transformations for beam pickup and delivery. RoboCrane then executes the movements (Figure 3).

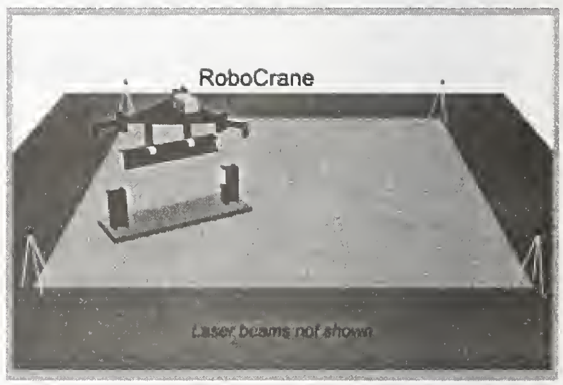


Figure 3: Steel beam placement with RoboCrane.

2.2 RoboCrane

RoboCrane is an innovative cable-driven manipulator invented by the NIST Intelligent Systems Division and further developed and adapted for specialized applications over a period of several years [2,3,4,5]. The basic RoboCrane is an inverted Stewart platform parallel link manipulator with cables and winches serving as the links and actuators, respectively. The moveable platform, or "lower triangle," is kinematically constrained by maintaining tension in all six cables that terminate in pairs at the vertices of the "upper triangle." This arrangement provides improved load stability over traditional lift systems and enables 6 DOF payload control.

The version of RoboCrane used in the ASCT project is the Tetrahedral Robotic Apparatus (TETRA) (Figure 4). In the TETRA configuration, all winches, amplifiers, and motor controllers are located on the moveable platform. The upper triangle only provides the three tie points for the TETRA cables, allowing the device to be retrofitted to existing overhead lift mechanisms.

Figure 4: RoboCrane – TETRA Configuration

2.3 Site Measurement System

The SMS uses commercially available positioning technology (*3D-I*) produced by Arc Second, Inc. in the *Constellation* and *Vulcan* product families. (*3D-I*, *Constellation* and *Vulcan* are registered trademarks of Arc Second, Inc.) These systems use stationary, active-beacon laser transmitters and mobile receivers to provide millimeter-level position data.

2.3.1 SMS Description

Both *Constellation* and *Vulcan* systems use eye-safe laser transmitters to triangulate the position of a tuned optical detector. Each transmitter emits two rotating, fanned laser beams and a timing pulse. Elevation is calculated from the time difference between fan strikes. Azimuth is referenced from the timing pulse. The field of view of each transmitter is approximately 290° in azimuth and +/- 30° in elevation/declination. The recommended minimum and maximum operating ranges from each transmitter are 5 m and 50 m, respectively.

Line-of-sight to at least two transmitters must be maintained to calculate position. The *Constellation* receivers each track up to four transmitters and wirelessly transmit timing information to a base computer for position calculation. The *Vulcan* system is a self-contained digitizing tool with two optical receivers on a rigid pole. A vector projection along the line formed by the two optical detectors allows 3-D measurement of the tool tip. *Vulcan* can track only two transmitters at one time; however, the transmitter selection can be manually switched between any of the four available. Recovery of positional data following momentary signal blockage takes approximately one second.

2.3.2 Prior *3D-I* / Mobile Robot Integration

Early efforts to use this laser technology for mobile robot navigation showed that although the system was capable of guiding a mobile robot [6], its use was restricted due to loss of track at relatively low vehicle speeds [7]. Upgrades to the positioning technology continued and a successful combination of indoor 2-D map creation and autonomous navigation was demonstrated in a research project at the Rochester Institute of Technology [8] (Figure 5). Subsequently, a single receiver *Constellation* system was installed on an autonomous lawn mower at the Carnegie Mellon University Field Robotics Center and provided positional reference in a large outdoor setting [9].

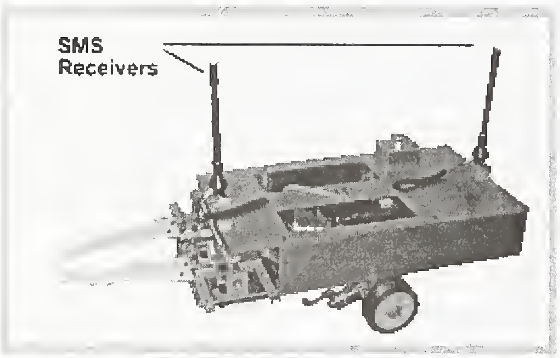


Figure 5: SMS Receivers on the RIT/SME "RC" Robot.

2.3.3 The SMS on RoboCrane

Three SMS receivers are mounted on RoboCrane at the vertices of the lower triangle (Figure 6). The receiver locations are registered to the manipulator during the initial setup process in the local SMS coordinate frame. For convenience, all measurements are calculated in the local SMS coordinate frame, though if required, mapping to an existing world coordinate frame could be accomplished. Receiver timing signals and diagnostic data are wirelessly transmitted to a base station computer running Arc Second's proprietary position calculation software. Position and SMS diagnostic information is polled at approximately 7 Hz using a NIST-developed data communications application. Position data from the three receivers are used to calculate RoboCrane's pose. Diagnostic data such as number of visible transmitters, excess signal noise or multipath reflections are also provided for each position calculation and is used to assess the quality of individual position fixes.

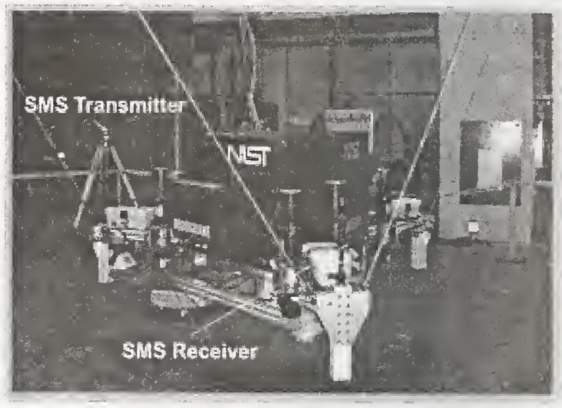


Figure 6: RoboCrane with SMS.

3.0 PERFORMANCE METRIC ANALYSIS OF THE DEMO III SENSOR SUITE

The NIST Intelligent Systems Division has provided research services developing control system architectures, advanced sensor systems, and standards to achieve autonomous mobility for various DOD unmanned ground vehicle programs including the Army Experimental Unmanned Ground Vehicle System (DEMO III) [10]. In a future experiment, high-resolution LADAR scanning and the SMS will be used to quantify the performance characteristics of the DEMO III sensor suite.

The test vehicle used by NIST is an Army HMMWV instrumented with a number of sensors (Figure 7). These include a LADAR range sensor that returns a 90° by 20° range image, a pair of color cameras for stereo imaging, a color camera, a line scan LADAR, and a wide-angle panoramic image mosaic obtained by stitching images from three color cameras. The vehicle also has an inertial navigation system (INS) and a GPS receiver.

An artifact field consisting of variable-sized box structures distributed within a nominal 30 m x 30 m area of clear terrain will be created on the

NIST grounds. CMAG will create a "ground-truth" digital map of the artifact field and then track the HMMWV test vehicle within that field.

Figure 7: NIST Robotic HMMWV Test Vehicle.

The artifact field will be scanned from several viewing angles using the using a high-resolution LADAR. The LADAR frame data will then be post-processed and manually meshed to create a multi-view map of the entire test area (Figure 8). The artifact field will also be measured using the SMS digitizing tool to create control points to overlay the SMS tracking data on the LADAR map.

Figure 8: Sample High-Resolution Multi-Scan LADAR Image (Traffic Intersection).

Three SMS receivers will be mounted on the HMMWV test vehicle. These receivers will be mounted high on the vehicle to ensure each receiver has line-of-sight to at least two of the four transmitters located on the perimeter of the test field during tracking. Fiducial points on the vehicle will be measured to map the SMS receiver locations to the vehicle coordinate system. The HMMWV test vehicle will then traverse the test field at slow speeds and the track data will be written at ~ 7 Hz. Track data from each receiver will then be post-processed to provide time-stamped vehicle pose (Figure 9).

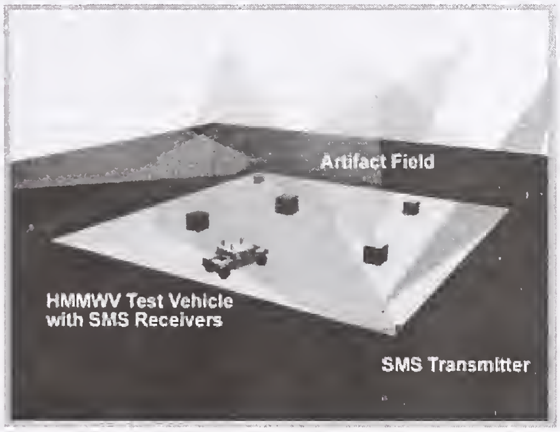


Figure 9: Test Vehicle on Artifact Field.

The 6 DOF vehicle track measured by the SMS will then be overlaid on the LADAR map. This will then be compared to the world map created by the DEMO III sensor suite to evaluate its performance. The performance of individual sensors and the performance of the system as a whole will be measured.

There are a number of measurement goals for the sensor characterization experiment. One is to see how accurately each of the sensors can measure the real terrain and the sizes, shapes, and positions of objects on it. Others are to see how well the sensors are registered and how well their positions in the vehicle's coordinate system can be determined. While the sizes of the objects within the test range can be measured accurately, the terrain is not entirely flat, and the high resolution LADAR will be used to determine ground truth. The SMS will be used to determine how well the INS system works.

Tables 1 through 5 provide manufacturer's performance data for the measurement systems that will be used in the test.

4.0 CONCLUSION

The combination of high-resolution LADAR imaging with a laser-based SMS provides an opportunity to create a "ground-truth" digital model of an autonomous vehicle's environment and then track the vehicle as it traverses, senses, and models that environment. The "ground-truth" map, complete with time-stamped vehicle pose, can then be compared to the platform's world model to quantify the performance of the on board navigation and sensing suite. A future experiment using the NIST HMMWV test vehicle on an artifact site will evaluate this method of mobile robot performance metric analysis.

5.0 REFERENCES

1. Lytle, A. M., Saidi, K. S. and Stone, W. C., "Development of a Robotic Structural Steel Placement System." *Proc., 19th International Symposium on Automation and Robotics in Construction*, Washington, D.C., September 23-25, 2002.
2. Stewart, D., "A Platform with Six Degrees of Freedom," Proc. of the Inst. of Mechanical Engineering, Volume 180(15), Part I:371-386, 1965-1966.
3. Albus, J. S., Bostelman, R. V., Dagalakis, N. G., "The NIST ROBOCRANE, A Robot Crane", Journal of Robotic Systems, July 1992.
4. Bostelman, R., Albus, J., Dagalakis, N., Jacoff, A., "RoboCrane Project: An Advanced Concept for Large Scale Manufacturing," *Proc, Association for Unmanned Vehicles Systems International*, Orlando, FL, July 1996.
5. Bostelman, R. , Albus, J., Stone, W., "Toward Next-Generation Construction Machines." *Proc. Of American Nuclear Society 9th International Topical Meeting on Robotics and Remote Systems*, Seattle, WA, March 4-8, 2001.
6. Beliveau, Y. J., Fithian, J. E., and Deisenroth, M. P., "Autonomous Vehicle Navigation with Real-Time 3-D Laser Based Positioning for Construction." Automation in Construction, (5) 1996.
7. Lytle, A. M., "Development of a Remotely Piloted Vehicle (RPV) with Real-Time Position Measurement (RtPM) for Hazardous Waste Site Characterization." Master's Thesis, Virginia Tech, Blacksburg, VA, 1994.
8. Rochester Institute of Technology, Society of Manufacturing Engineers (Student Chapter 18), Robot Construction Project, 2001. www.rit.edu/~smewww/project/rc/rc.html
9. Batavia P. H., Roth S. A., Singh S., "Autonomous Coverage Operations In Semi-Structured Outdoor Environments." *Proc., IROS*, Lausanne, Oct. 2002.
10. Shoemaker, C. M. and Bornstein, J. A., "The Demo3 UGV Program: A Testbed for Autonomous Navigation Research." *Proc., IEEE International Symposium on Intelligent Control.* Gaithersburg, MD 1998.

Table 1: GDRS Area-Scan Ladar Specifications (HMMWV Test Vehicle).

Property	Specification
8 laser beams, 1 rotating mirror	With 8 facets
Scan resolution	32 lines x 180 pixels
Scan coverage	20 x 90
Angular resolution	0.658 x 0.5
Maximum frame rate	60 scans/s but 30 scans/s

Range	5 m to 70m (vertical surface)
Range resolution/standard uncertainty	±7.6 cm / 15 cm
Data measurement rate	Range: 345,600 measurements/s
Day/Night Operation	Range Independent of ambient light

Table 2: Real-Time Performance of Applanix POS LV 420 Intertial Navigation Unit (HMMWV Test Vehicle).

POS LV 420-RT (using DGPS)	**GPS Outage Duration (minutes)**					
	0 min	**1 min**	**3 min**	**5 min**	**10 min**	**20 min**
X, Y Position (m)	1.0	1.5	1.75	2.0	2.5	3.5
Z Vertical Position (m)	1.5 to 2.0	2.0	2.0	2.0	2.5	3.0
Roll & Pitch (deg)	0.02	0.02	0.02	0.02	0.02	0.02
True Heading (deg) 0.02	0.02	0.02	0.04	0.06	0.10	0.20

Table 3: Performance of Applanix LV 420 with post-processing (HMMWV Test Vehicle).

POS LV 420 (post processed)	**GPS Outage Duration (minutes)**					
	0 min	**1 min**	**3 min**	**5 min**	**10 min**	**20 min**
X, Y Position (m)	0.02	0.12	0.40	0.75	1.5	2.5
Z Vertical Position (m)	0.03	0.15	0.50	0.65	1.0	2.0
Roll & Pitch (deg)	0.005	0.005	0.005	0.007	0.007	0.09
True Heading (deg)	0.02	0.02	0.02	0.03	0.035	0.035

Table 4: Manufacturer's Specifications - Riegl LMS Z210 Ladar.

Property	**Specification**
Scan coverage	80° x 330°
Angular stepwidth	0.072° to 0.36°
Angular readout accuracy	0.036°
Frame scan rate	1 °/sec to 15 °/sec
Minimum Range	2 m
Maximum Range	350 m (25 mm resolution, natural target)
Range resolution	25 mm or 50 mm, selectable
Standard uncertainty	Resolution + Distance error of = ±20 ppm

Table 5: Manufacturer's Specifications - *Constellation* and *Vulcan*.

Property	**Specification**	
	Vulcan	*Constellation*
Transmitter coverage	60° x 290°	60° x 290°
Transmitters required for position calculation	2	2
Maximum number of observable transmitters	2	4
Nominal Laser rotation rate	40 Hz to 50 Hz	40 Hz to 50 Hz
Minimum Range	5 m	5 m
Maximum Range	50 m	35 m
Angular resolution	=90 μrad	=90 μrad
Data rate (position calculation)	10 Hz	7 Hz

Standard uncertainty (Instrument x,y,z – U66)	± (1 + D/10000) [mm] D (mm) - distance between transmitters	10*RSS (250, D*8) [µm] RSS- Root Sum Squares RSS(A,B) = $\{A^2+B^2\}^{1/2}$ D (m) –distance from farthest transmitter No point data averaging

DISCLAIMER: Certain commercial equipment, instruments, or materials are identified in this report in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

An Uncertainty Propagation Architecture for the Localization Problem

Arnaud CLERENTIN, Laurent DELAHOCHE, Eric BRASSART, Cyril CAUCHOIS
CREA - Centre de Robotique, d'Electrotechnique et d'Automatique
IUT, département Informatique, Avenue des Facultés, 80000 Amiens – France
{Arnaud.Clerentin, Laurent.Delahoche}@u-picardie.fr

ABSTRACT

In this article, a dynamic localization method based on multi target tracking is presented. The originality of this method is its capability to manage and propagate uncertainties during the localization process. This multi-level uncertainty propagation stage is based on the use of the Dempster-Shafer theory. The perception system we use is composed of an omnidirectional vision system and a panoramic range finder. It enables to treat complementary and redundant data and thus to construct a robust sensorial model which integrates an important number of significant primitives. Based on this model, we treat the problem of maintaining a matching and propagating uncertainties on each matched primitive in order to obtain a global uncertainty about the robot configuration.

KEYWORDS : *mobile robot localization, omnidirectional vision, uncertainty management, multi target tracking*

1 INTRODUCTION

Localization is a fundamental problem in mobile robotics. Mobile robots have to be able to locate themselves in their environment in order to accomplish their tasks. In order to act in a robust way and to increase the reliability in operation, the robot should consider data as uncertain and all decision should be made using data of an appropriate level of certainty. The localization method presented in this paper has the particularity to integrate uncertainty quantification and to propagate low-level data uncertainties along the localization process. The goal is to obtain a global uncertainty about the robot localization. In this purpose, we propose an architecture which allows to manage and propagate uncertainty. The Dempster-Shafer theory [8] is the key tool of this architecture. Indeed, this formalism enables to easily treat uncertainty since it permits to attribute mass not only on single hypothesis, but also on union of hypothesis. We can thus express ignorance. This is the main difference with Bayesian theory.

Localization methods can be classified as being relative (based on the use of proprioceptive data) or absolute (based on the use of exteroceptive data). Absolute methods consist in determining the robot's position with the only use of exteroceptive data: the robot's configuration is calculated in the environment reference without using previous information [1][5]. But the problem of this kind of localization is linked to the matching stage between the sensorial model and the theoretical map of the environment: this stage can be highly combinative and non robust in connection with multiple solutions, for example with symmetrical environments. In order to increase the reliability and decrease the computation time of these methods, the use of multi target tracking can be interesting. In the case of the localization problem, multi target tracking can be seen as a propagation of an initial matching. This paradigm is abundantly treated is the literature, for example by Bar Shalom [4]. The methods generally used are probabilistic ones and the two main are JPDAF (Joint Probabilistic Data Association Filter) [4] and MHT (Multiple Hypothesis Tracker) [3]. But these two methods have some drawbacks. They need to know the false alarm rate. The JPDAF takes into account a fixed number of targets and doesn't initialize new tracks. The MHT has combinatorial problems. Therefore, we propose in this paper a multi target tracking method for the localization problem based on the Dempster-Shafer theory used in a framework called *extended open world* [7]. Since this method uses DS theory, it naturally integrates our uncertainty propagation architecture and enables to manage an uncertainty for each target. It allows also to treat the problem of target apparition and momentarily disappearance.

This paper is organized as follow. In a first part, we present our perception system. Then we deal with the target classification stage based on the exploitation of the complementary and redundancy of the data provided by our perception system. Section 4 explains our target tracking algorithm. The paper ends with experimental results presentation.

2 THE OMNIDIRECTIONAL PERCEPTION SYSTEM

Our original perception system uses two omnidirectional sensors in cooperation: the omnidirectional vision system SYCLOP and a panoramic range finder system [10] (Fig. 1).

These two sensors have been developed and used independently within our laboratory. The range finder system is an active vision sensor [10]. It allows to obtain a robust omnidirectional range finding sensorial model. The interest of this system is on the one hand its low cost and on the other hand its robustness facing a high incidence angle. The SYCLOP system [2], similar to the COPIS one [14], is composed of a conic mirror and a CCD camera. It enables us to get radial straight lines which characterize angles of every vertical object such as, for example, doors, corners, edges (Figure 2)...

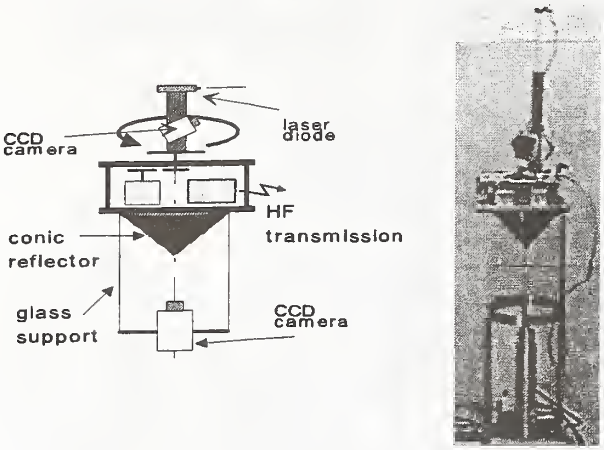


Figure 1: The perception system and the prototype we built.

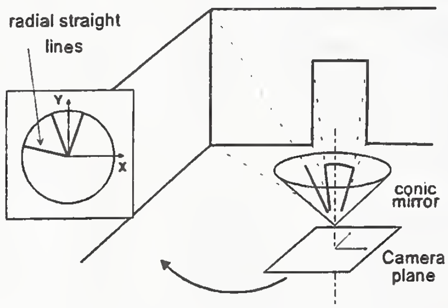


Figure 2: Principle of the omnidirectional sensor SYCLOP

This two omnidirectional sensors association is interesting since it permits to manage some complementary and redundant information within the same sensorial model :

- With SYCLOP, the radial straight lines give the angular position of every vertical object, but the information of depth cannot be achieved in one acquisition: it is not possible to differentiate with this only sensor the notion of opening (corridor, opening of door...) and the notion of vertical object (closed door, radiator,...) (Figure 3).
- The range-finder system, following a segmentation stage [10], permits to exploit sensorial primitives that are segments. In this case we have the notion of depth, but it is impossible to differentiate two vertical objects placed in the same alignment: two closed doors placed on the same wall. It misses the notion of angle that will be provided by the SYCLOP system.

So this association enables to construct a highly descriptive sensorial model, richer than the models obtained with each sensor individually (Figure 3).

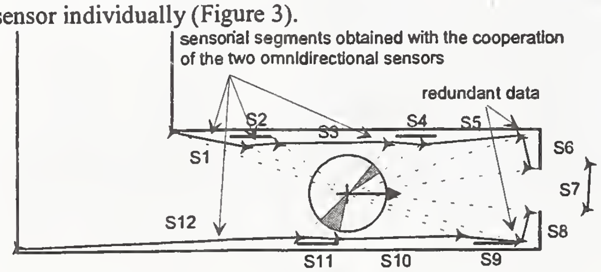


Figure 3: Principle of the omnidirectional sensorial cooperation.

3 SENSORIAL MODEL CONSTRUCTION

3.1- Segment primitives determination and associated uncertainty computation

The final primitives of the sensorial model are segments. They are determined with two types of approaches [6]:

- data complementarily approach. The first case concerns the data detected by SYCLOP but not by the depth sensor. In this case, the treatment consists in cutting up segments gotten with the range finder in subsegments according to the radial straight lines of the vision system (case 2 of Figure 4). The second case concerns the data detected by the depth sensor but not by SYCLOP. In this case, the breakpoint gotten by the Duda Hart segmentation method is directly considered (case 3 of Figure 4).
- data redundancy approach. The redundant aspect is characterized by the detection of a vertical landmark with the two sensors (Figure 4). In this case, we use the radial straight line to determine the segment endpoint. Indeed, we consider that the SYCLOP sensor has a better angular precision.

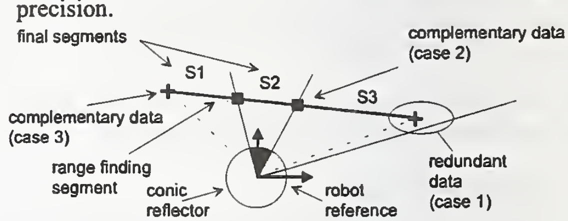


Figure 4: The different cases of the cooperation algorithm.

After the determination of the sensorial model, we compute the reliability, i.e. the uncertainty of each segment. This stage is preponderant for the multi-target tracking stage presented in this article. In this purpose, we take into account five criteria.

The first criteria is the mean distance between the range finding points contained by the segment and this segment. If this mean distance is high, it means that the points are not very well aligned, so this segment is not very sure.

The second criteria is the number of points supported by the segment. This criteria is only discriminative when the segment contains very few points. In this case, it is not very sure.

The third criteria is the segment density of points. As shown in [10], a major drawback of this kind of triangulation depth sensor is a decreasing resolution with an increasing distance. So, this criteria, which is linked to the mean distance between the sensor and the set of point, is a good indicator of the segment reliability (more distant the set of points is, less the precision is).

The fourth criteria analyzes if the segment is detected by one or by the two sensors. The worst case occurs when the two extremities of the segment are detected only by the laser range finder (case 1 of Figure 5). The best case occurs when the two extremities are detected by the two sensors (case 5 of Figure 5). Between these extreme cases, we can distinguish three others cases [6].

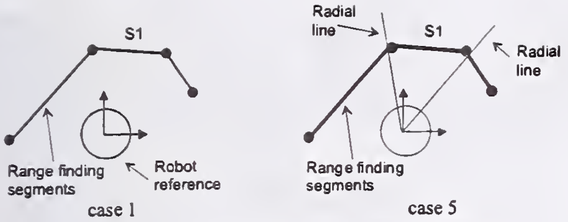


Figure 5 : the two extreme cases of the fourth criteria.

The last criteria concerns a gray level curve extracted from the SYCLOP image. We take into consideration five concentric gray level circles whose average is made. We obtain thus one gray level curve from 0 to 360 degrees. We apply on the portions of curve which represent a segment a least square algorithm. We obtain a straight line and we compute the mean difference of the gray level values from this line. If the difference is high, this means that the gray level sector is not constant. This case occurs generally when a landmark has not been detected by SYCLOP, so this segment is not sure (Figure 6).

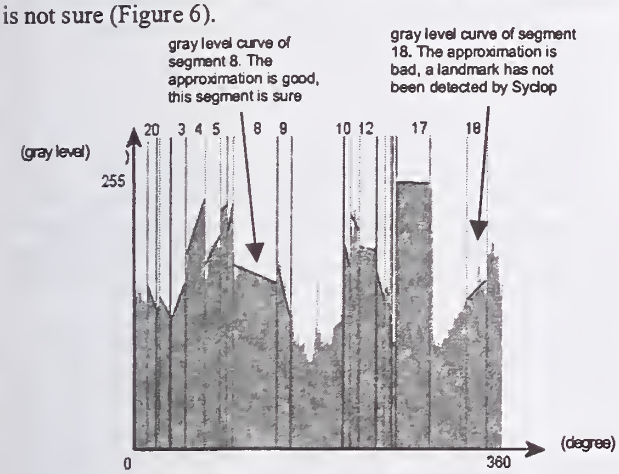


Figure 6: an example of gray level curve concerning the fifth criteria.

The fusion of these five criteria is made thanks to the Dempster-Shafer theory [8]. Our frame of discernment (FOD) is composed of two elements: "YES" and "NO" corresponding to the assertions "The segment exists" and "the segment does not exist". We show on Figure 7 one of the five BPAs which integrates the ignorance quantification. The Dempster rule of combination [8] gives $m_{seg}(YES)$, $m_{seg}(NO)$ and $m_{seg}(\Theta)$. The segment uncertainty is denoted by this set mass m_{seg}. We have studied on 50 experimental sensed map the conflict between these five criteria. Indeed, these five criteria are redundant and conflict can arise. Experimentally, we have noticed that it is not important (mean conflict = 0.13). This shows that these criteria are pertinent and lead to a consensual decision. But, in certain cases, the conflict is high. So, we have decided to work in an *open world* context [12], i.e. not to normalize. Indeed, in case of high conflict, as Zadeh showed, a normalization can lead to an aberration. On the other hand, a non normalization gives us a precious indication about the conflict between the five criteria. So we report the conflict (the mass on ∅) to the ignorance Θ.

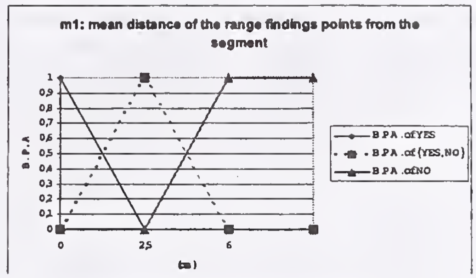


Figure 7: B.P.A of the first classification criterion ({YES,NO}=Θ)

3.2- High level primitives determination

The next stage consists in determining high semantic level primitives which are: "corner", "edges", "wall" and "other" (Figure 8). The "other" class characterizes landmarks which are not "corner", "edges", "wall".

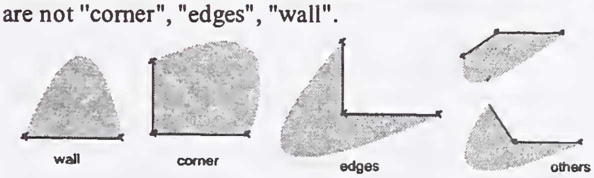


Figure 8: High level semantic primitives.

We use the high semantic level entities "corner", "edge" and "other" because the azimuth angle of the junction of the two segments is a "strong" angle (important existence probability): it is a discriminating angle in connection with the occultation problem. The angles of a segment primitive can be false angle due to occultation.

As in the previous step, we compute an uncertainty linked to each primitive. This uncertainty is determined by propagating the segment(s) uncertainty(ies) computed on the previous step. We reach this aim in two stages. Firstly, we determine the type of the primitive. Secondly, we compute its uncertainty.

We determine the primitive type by fusing two criteria (Figure 9). The first criteria m_1 is the angle α between two consecutive segments S1 and S2 of the sensorial model. The second criteria m_2 is the minimal distance *d* between the "junction" extremities of the two segments S1 and S2. The belief functions of these two criteria are discussed on [9].

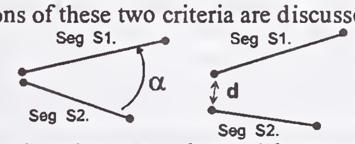


Figure 9: angle criteria and minimal distance criteria.

The fusion is made thanks to the Dempster rule of combination and enables to obtain the mass set m_{type} by fusing m_1 and m_2. The two criteria taken into account are complementary, so there is no conflict. The taken decision is the one which has the maximal credibility.

The second stage consists in computing the high level primitive uncertainty. In this purpose, we take into account two uncertainties:

- the uncertainty of the segment(s) composing the primitive
- the uncertainty on the primitive type computed on the first stage.

The FOD is composed of two elements: YES and NO corresponding to the assertions "YES, the primitive exists" and "NO, the primitive does not exist". The first criteria m_{1prim} is linked to the segment uncertainty coefficient m_{seg}.

For a primitive wall: $$\begin{cases} m_{1prim}(YES) = m_{seg}(YES) \\ m_{1prim}(NO) = m_{seg}(NO) \\ m_{1prim}(\Theta) = m_{seg}(\Theta) \end{cases}$$

For a primitive corner, edge or other composed of two segments S1 et S2:

$$\begin{cases} m_{1prim}(YES) = m_{seg}^{S1}(YES) \oplus m_{seg}^{S2}(YES) \\ m_{1prim}(NO) = m_{seg}^{S1}(NO) \oplus m_{seg}^{S2}(NO) \\ m_{1prim}(\Theta) = m_{seg}^{S1}(\Theta) \oplus m_{seg}^{S2}(\Theta) \end{cases}$$

The second criteria m_{2prim} for a primitive of type T is computed according to the following rules:

$$\begin{cases} m_{2prim}(YES) = Cr(T) \\ m_{2prim}(NO) = \sum_{A \subseteq \Theta, A \cap T = \emptyset} m_{type}(A) = Cr(\overline{T}) \\ m_{2prim}(\Theta) = \sum_{A \subseteq \Theta, A \neq T, A \cap T \neq \emptyset} m_{type}(A) = Pl(T) - Cr(T) \end{cases}$$

The mass for the YES is equal to the belief we have on T, i.e. the credibility of T Cr(T). The mass for the NO is equal to the disbelief on T, i.e. the mass which is not on T. The mass for Θ represents the uncertainty about T, i.e. the mass which is on focal elements which include T. Doing this, we respect the constraint that the mass sum must be equal to 1.

By fusing the two criteria m_{1prim} et m_{2prim}, we obtain the uncertainty of the primitive through m_{prim}(YES), m_{prim}(NO) and $m_{prim}(\Theta)$. Doing this, we estimate the uncertainty of the high level primitives by propagation of the segments uncertainty.

So, at the end of this step, we have four lists of primitives (a list of corners, of edges, etc.) with an associated uncertainty for each primitive through the set mass m_{prim}. This uncertainty includes the uncertainty about the type of the primitive and the uncertainty about the existence (the reliability) of the segments which compose the primitive.

4 DYNAMIC LOCALIZATION METHOD

4.1 - Algorithm

Our localization method is based on a tracking of high semantic level primitives: we propagate the matching made at an acquisition n on an acquisition n+1. So, the problem to solve is the following: propagation of an initial matching on the acquisitions realized during the robot's displacement. The initial matching is done in manual way or with the absolute localization method presented in [6]. Then we try to pursue the matching. To confirm a matching propagation, we must before generate a prediction which will be compared to the observations. So we have developed a prediction system based on a linear extrapolation of the azimuth angle curves of the high level primitives (on experimental results, we can note that the angles variation is locally linear): we generate a predictive observation vector composed of angles got by linear extrapolation (Figure 11). For example, if we examine the evolution of the landmark angles $\Theta1$, $\Theta2$ and $\Theta3$ (Figure 10), we remark that the curve can be extrapolated in order to have a prediction $\Theta4_p$. If a matching is done between $\Theta4_p$ and an angle observation, the track is propagated.

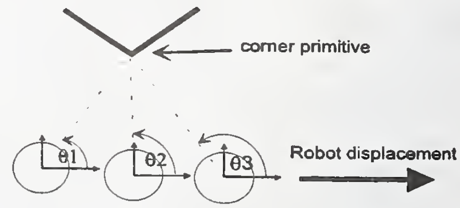


Figure 10: evolution of landmark angles.

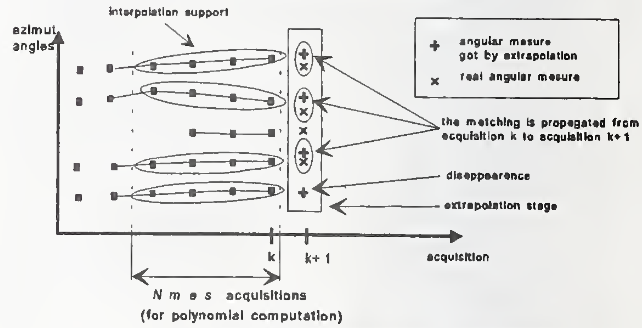


Figure 11: principle of angular measures extrapolation.

Our prediction heuristic is robust since it is based on angle curves of high level primitives: the extrapolated measures correspond to "strong" angles whose evolution curves can not confuse themselves because they do not suffer of occultation phenomena.

At this level, the problem is to match for each type of primitive the p angular observations obtained at the acquisition t with the q predictions. These q predictions are computed from the *Nmes* last observations. To reach this aim, we use the Dempster-Shafer theory in the framework of *extended open word* [7] because of the introduction in the FOD of an element noted * which represents all the hypothesis which are not modeled.

For each prediction *Qj* ($j \in [1,q]$), we apply the following algorithm.

- The frame of discernment Θ is composed of:
  - the p observations (*Pi* means "the prediction *Qj* is matched with the observation *Pi*")
  - and the element * which means "the prediction *Qj* cannot be matched with one of the p observations".

  So: $\Theta = \{P_1, P_2, ... P_p, *\}$
- The matching criterion is the angular difference between observation *Pi* and prediction *Qj* (Figure 11).
- For each observation *Pi*, we compute :
  - $m_i(P_i)$ the mass associated with the proposition "*Pi* is matched with *Qj*".
  - $m_i(\overline{P_i})$ the mass associated with the proposition "*Pi* is not matched with *Qj*".
  - $m_i(\Theta_i)$ the mass represented the ignorance concerning the observation *Pi*.

The BPAs are shown on Figure 12.

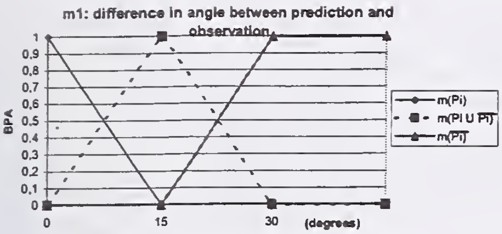


Figure 12: BPA of the matching criterion.

After the treatment of all the Pi observations, we have p triplets :

$$m_1(P_1) \quad m_1(\overline{P}_1) \quad m_1(\Theta_1)$$
$$m_2(P_2) \quad m_2(\overline{P}_2) \quad m_2(\Theta_2)$$
$$\ldots$$
$$m_p(P_p) \quad m_p(\overline{P}_p) \quad m_p(\Theta_p)$$

We fuse these triplets and we get $m_{match}(P_1)$, $m_{match}(P_2)$,..., $m_{match}(Pp)$, $m_{match}(*)$ and $m_{match}(\Theta)$ by using the condensed formulas obtained by Gruyer in [11].

- The final decision is the one which has the maximal BPA.

Experimentally we can note that ambiguities can appear after this step, but only on the segment primitives: a segment observation Pi can be matched with two segment predictions Qi, this case is impossible in the reality. So we use, like Gruyer, only for this class of primitives, a traditional assignment Hungarian algorithm to match one observation with one prediction [11].

Finally, this matching method enables us to easily manage primitive appearances and disappearances:

- If an element Pi of the FOD cannot be matched, Pi is an appeared primitive and a track can be initialized.
- If a prediction Qj is matched with *, the track is temporarily or definitively lost.

4.2 - Management of an appearance

From the position computed with the matched primitives, we try to match the appeared primitives with the primitives of the theoretical map which is composed of four lists (a list of wall, a list of corner, etc.). In other words, we try to initiate a new track. We have to distinguish two cases: primitives wall and the other primitives.

For each appeared primitive wall, we have considered three correspondence tests applied on all the theoretical wall primitives [13] :

- angular difference α between the two segments,
- difference in length (Ls-Lm) between the two segments,
- distance D between the centers of the two segments.

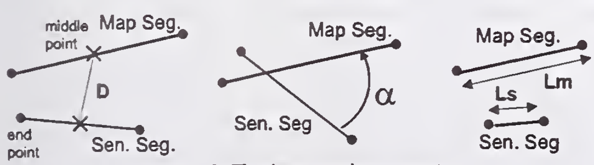


Figure 13: The three matching criteria.

The fusion of these three treatments is made thanks to the Dempster-Shafer theory. Our FOD is composed of two elements: YES and NO corresponding to those assertions : "Yes, we can match the two walls" and "No, we can not match the two walls". For each criterion, we have determined the BPAs m_1, m_2, m_3 (see Figure 14 for an example of BPA).

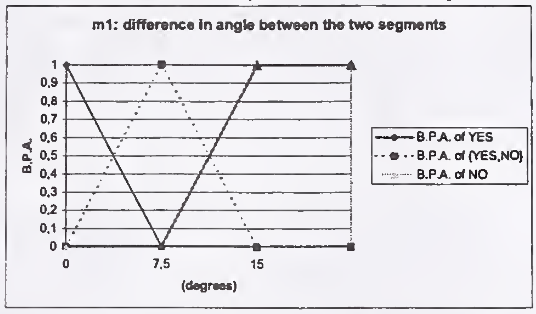


Figure 14: :basic probability assignments of the first matching criteria ({YES,NO}=Θ)

We can then perform the combination calculation thanks to the Dempster-Shafer rules without renormalization [12] in order to get a mass set m_m. The non-renormalization gives us a precious indication about the conflict. Generally, we have experimentally noticed that this conflict is null, but, in a few cases, it can be high. This occurs for example when we examine two parallel walls. So, if the conflict k is superior to 0.7, we think this value is too high and we take a prudent decision: we don't match the two segments. If k<0.7, we can take a decision and the segments are matched if BPA for the *YES* $m_m(YES)$ is superior to the BPA for the *NO* $m_m(NO)$.

For each other primitive (corner, edge, other), we consider two correspondence tests (Figure 15):

- The difference between the robot-sensorial primitive distance d_{seg} and the robot-map primitive distance d_{map}.
- The difference between the sensed primitives angle Θ_{seg} and the theoretical primitive angle Θ_{map}.

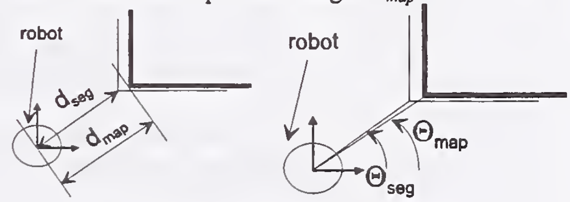


Figure 15: The two matching criteria

As the previous case, our FOD is composed of two elements: YES and NO. The fusion is realized according to the same strategy as the wall primitives.

4.3 - Management of a disappearance

As we will see in paragraph 4.4, if a matching is not propagated, the track is not immediately cancelled but its uncertainty increases. If this uncertainty becomes too high, we definitively cancel this track.

4.4 - Track uncertainty management

For each track, we manage an associated uncertainty with the help of the Dempster-Shafer theory. Our FOD for each track is composed of two elements: "YES" and "NO" which mean "Yes, the track exists" and "No, the track does not exist". Two stages are managed:

Uncertainty initialization stage. In the case of a primitive appearance, the initial uncertainty $m_{track\ 0}$ at time 0 takes into account the uncertainty of the primitive m_{prim} (paragraph 3.2) and the uncertainty of the first matching m_m (paragraph 4.2).

So, the two criteria are:

- $\begin{cases} m_1(YES) = m_{prim}(YES) \\ m_1(NO) = m_{prim}(NO) \\ m_1(\Theta) = m_{prim}(\Theta) \end{cases}$

  m_1 takes into account the uncertainty of the primitive.
- m_2 which takes into account the uncertainty of the first matching through m_m(YES) [9].

We have noticed on experimental results that conflict can appear, but it occurs in only one case: a good matching of an unreliable primitive. Our strategy to manage this conflict is to reduce the weight of the primitive uncertainty m_1 by an operation of discounting [8]:

$$\text{if } m_1(\text{YES})>0 : \begin{cases} m_1^a(YES) = m_1(YES)\times(1-m_1(NO)) \\ m_1^a(NO) = m_1(NO)\times(1-m_1(NO)) \\ m_1^a(\Theta) = 1 - m_1^a(YES) - m_1^a(NO) \end{cases}$$

We obtain $m_{track\ 0}$ (YES), $m_{track\ 0}$ (NO) *and* $m_{track\ 0}$ (Θ) by merging m^{α}_1 and m_2 using the Dempster combination rule.

If $m_{track\ 0}$ (NO) > $m_{track\ 0}$ (YES), then we consider that the uncertainty is too high and we don't initialize the track. This taking into account of the primitive uncertainty enables us not to work with all the primitives, we privilege the "robust" and reliable primitives.

Uncertainty propagation stage. Then, if the matching can be propagated, the track uncertainty is updated by taking into account:

- In relation with time *t-1*: the track uncertainty at time *t*-1
- In relation with time *t*: the primitive uncertainty and the matching uncertainty at time *t*.

Let be $m_{track\ t-1}$ the mass set of the track at time *t*-1. The three set masses m_1,m_2 and m_3 concerning the 3 criteria are:

- $\begin{cases} m_1(YES) = m_{prim}(YES) \\ m_1(NO) = m_{prim}(NO) \\ m_1(\Theta) = m_{prim}(\Theta) \end{cases}$ m_1 takes into account the primitive uncertainty at time *t*.
- m_2 takes into account the uncertainty of the matching at time *t* m_{match} computed on paragraph 4.1 [9].
- $\begin{cases} m_3(YES) = m_{track\,t-1}(YES) \\ m_3(NO) = m_{track\,t-1}(NO) \\ m_3(\Theta) = m_{track\,t-1}(\Theta) \end{cases}$ m_3 takes into account the track uncertainty at time *t*-1.

We adopt the strategy described in the uncertainty initialization stage: a high conflict only appears when we realize a good matching of an unreliable primitive and we discount the mass of the primitive uncertainty.

$m_{track\ t}$(YES), $m_{track\ t}$(NO) and $m_{track\ t}$(Θ) are obtained by fusing m_1,m_2 and m_3.

If the matching is not propagated, the uncertainty of the track increases. In this case, we fix the BPA m_{match} as follow:

$$\begin{cases} m_{match}(YES) = 0 \\ m_{match}(NO) = 0.2 \\ m_{match}(\Theta) = 0.8 \end{cases}$$

This mass set has been determined experimentally in order to obtain a regular increase of the track uncertainty. So, if $m_{track\ t-1}$ is the BPAs of the track at time *t*-1, we update the uncertainty $m_{track\ t}$ using the Dempster rule of combination with $m_{track\ t-1}$ and m_{match}.

If $m_{track\ t}$(NO)>$m_{track\ t}$(YES), then we consider that the track uncertainty is too high and the track is definitively lost. Here again, the taking into account of the primitive uncertainty enables us to privilege the tracks with reliable primitives.

Transition between a primitive wall and an other primitive. We manage in our system the transition between the primitives corner, edge, other and the primitives wall. An example of such transition is shown on Figure 16. At time *t*, the robot detects one of the two faces of the edge and this face is classified as a wall primitive. At time *t*+1, the two edge faces are visible from the robot and it detects an edge primitive. The wall detected at time *t* and the edge detected at time *t*+1 correspond to the same track. So we use the uncertainty of the wall track at time *t* to initiate the uncertainty of the edge track at time *t*+1.

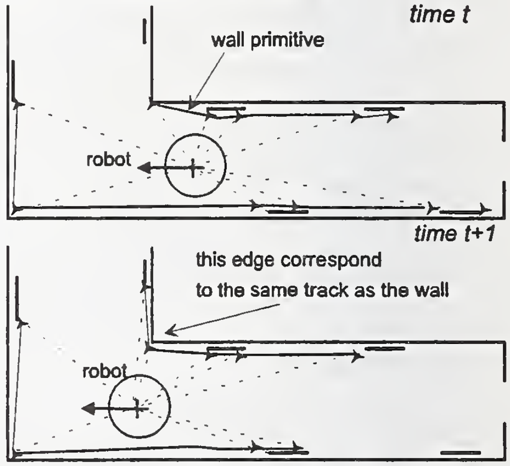


Figure 16: an example of transition wall → edge

4.5 - Localization uncertainty

The last step of our uncertainty propagation architecture is to compute the uncertainty of the robot localization. This aim is reached with the help of the Demspter Shafer theory and the FOD is composed of the two elements YES and NO corresponding to the assertions "Yes, the localization is correct" and "No, the localization is not correct". We take into account *p*+2 criteria.

The first criterion is the number of high level primitives used to localize the robot. Indeed, if we use few primitives, the localization is not reliable.

The second criterion is a ratio concerning the number of detected primitives and the number of matched primitives. Indeed, if we detect a lot of primitives but if we match only a little few primitives, this can mean that a problem occurs in the classification process or in the matching process. So the localization may be unreliable.

$$ratio = \frac{number\ of\ matched\ primitives}{number\ of\ detected\ primitives}$$

The last *p* criteria are the uncertainty of the *p* tracks managed by the robot, i.e. the *p* mass sets $m_{track\ t}$ computed in the paragraph 4.4. If the tracks are uncertain, the localization will be uncertain. Since we merge an important number of mass sets and since the Dempster operator is not idempotent, we apply an operation of discounting on the p mass sets $m_{track\ t}$. The discounting coefficient is different if the mass set $m_{track\ t}$ concerns a wall primitive or an other primitive (corner, edge and other): we privilege in the fusion process the "strong" primitives corner, edge and other.

These p+2 criteria are fused according to the Dempster rule and we obtain a mass set m_l which quantifies the localization uncertainty. This uncertainty is directly issued of the uncertainties of the low-level data which have been propagated, as shown on Figure 17.

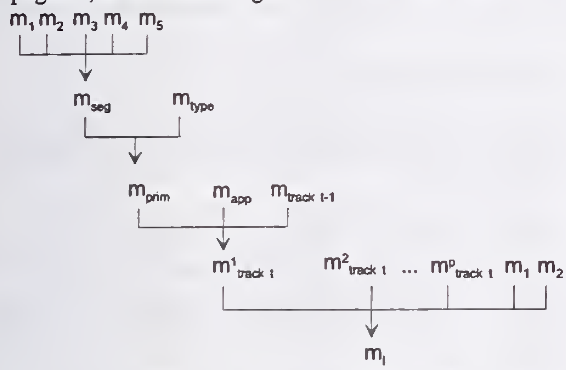


Figure 17 : Uncertainties propagation during the localization process.

5 EXPERIMENTAL RESULTS

We have tested our algorithm on several acquisitions made in an indoor environment (the end of a corridor shown Figure 18 whose theoretical map in possession of the robot is on Figure 19). The omnidirectional acquisitions and the localization algorithm are computed in a Pentium PC located on our mobile robot.

On Figure 18, we show an example of high level primitives sensed map. We report on Table 1 the different masses about the primitives uncertainty.

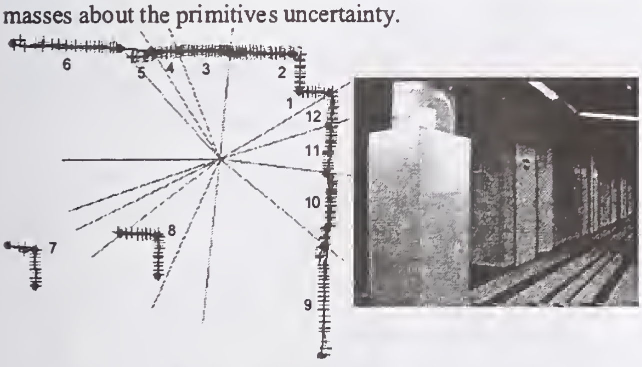


Figure 18: high level primitive map and the real environment.

On a two paths made in the corridor by our robot mobile SARAH, we can note on 42 acquisitions made every 30 cm that the robot's position is determined correctly with a good precision: the mean error is equal to 13 cm in position and 3 degrees in orientation (Figure 19).

On Figure 18, we represent only the tracked landmarks of the second trajectory. We can remark that our tracking is robust and efficient: among all the important number sensorial primitives, the tracked primitives are correctly identified and the tracks are generally never lost until the landmarks become invisible from the robot. We show on Figure 21 the uncertainty evolution of edge 6. The initial matching is done manually and the mass set is set as follow: $m_{track\ 0}$(YES)= $m_{track\ 0}(\Theta)$=0.5, $m_{track\ 0}$(NO)=0. The landmark is tracked until acquisition 7, so the BPA for YES $m_{track\ t}$(YES) increases. Then, it becomes invisible from the robot. So the BPA for YES decreases until acquisition 12 where the BPA for NO is superior to the BPA for YES. So the track is definitively lost.

Primitive number	Type	m(YES)	m(NO)	m(θ)
1	Edge	0.72	0.08	0.20
2	Corner	0.64	0.16	0.20
3	Wall	0.64	0	0.36
4	Wall	0.91	0	0.09
5	Wall	0.07	0	0.93
6	Wall	0.83	0	0.17
7	Edge	0.33	0.25	0.42
8	Edge	0.70	0.03	0.27
9	Wall	0.50	0	0.50
10	Wall	0.11	0	0.89
11	Wall	0.47	0	0.45
12	Corner	0.78	0.03	0.18

Table 1: uncertainties of the primitive model.

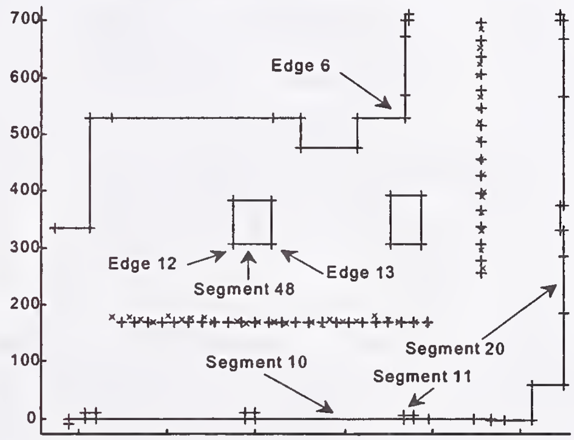


Figure 19: theoretical map and localization results ('+'=real position, 'x'=computed position).

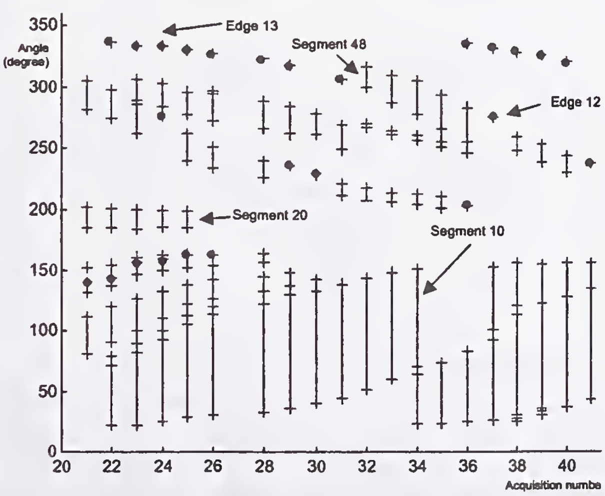


Figure 20: the tracked landmarks ('+'=corner,point=edge, segment=wall)

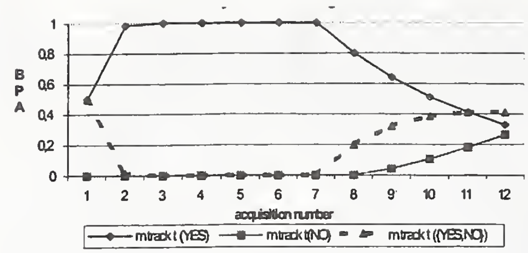


Figure 21: uncertainty evolution of landmark edge 6.

Finally, we see on Figure 16 an example of a double transition edge-segment-edge. Until acquisition 32, the two faces of the edge are visible. On acquisition 33, one face is visible, so a primitive segment is detected but we don't initiate a new track since this segment belong to the edge previously tracked. On acquisition 37, the robot can detect a new edge (edge 12 on Figure 19) that contains the segment. As the previous case, we don't initiate a new track but we prolong the current track.

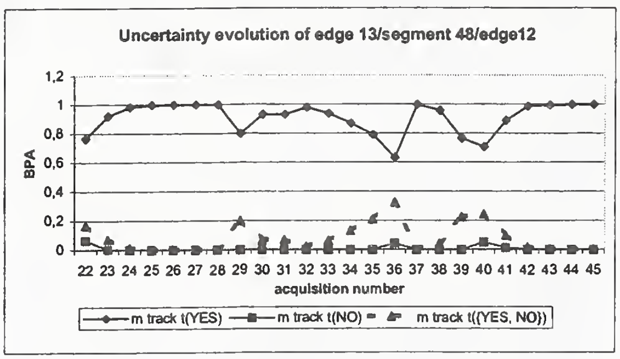


Figure 22: uncertainty evolution of edge 13/segment 48/edge 12.

On Figure 23, we show the evolution of the localization uncertainty. The uncertainties of the first acquisitions are weak : the number of tracked primitives is high. Then this number decreases, so the uncertainty increases, i.e. m_t(YES) decreases. After acquisition 46, several new tracks are initialized and the uncertainty becomes weak.

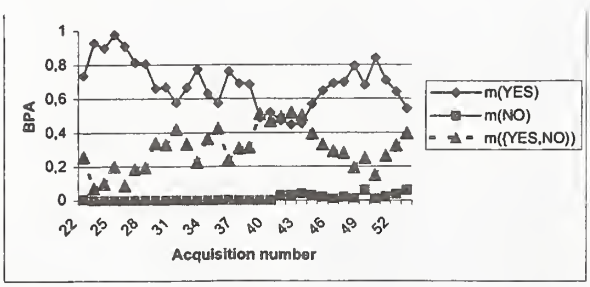


Figure 23 : localization uncertainty.

6 CONCLUSION

In this paper, we have studied and implemented a multi level uncertainty propagation architecture. After a multi criteria fusion stage based on the use of the Dempster-Shafer theory, we obtain a multi-valued sensorial map which permits to quantify the credibility of the high level primitives. These primitives are then used in our dynamic localization method based on a propagation of an initial matching This method solves two problems linked to the multi target tracking: the propagation of an uncertainty concerning the landmark tracks and the treatment of the apparition and momentary disappearance of a track. This multi-target tracking paradigm has been tested on several robot's path in a large structured indoor environment and has provided good results concerning the matching maintaining and the preciseness of the localization. An extension of this work could concern the linear angular prediction which is mono criteria. A prediction based on a dynamic model or combining 'proprioceptive' could be used and would allow the system to operate on fast moving vehicles.

7 REFERENCES

[1] S. Thrun, "Finding Landmarks for Mobile Robot Navigation", *IEEE Int. Conf. on Robotics and Automation*, May 1998, pp. 958-963.

[2] L. Delahoche, B. Marhic, C. Pégard, P. Vasseur - "A navigation system based on an omnidirectional vision sensor", Proc. of *Int. Conf. on Intelligent Robots and Systems*, France, Sept. 1997.

[3] D.B. Reid, "An algorithm for tracking multiple target", *IEEE trans. on automatic control*, vol. AC24, n. 6, pp. 843-844, Dec. 89

[4] Y. Bar-Shalom, « *Multitarget-Multisensor tracking : Applications and Advances vol.II.* », Artech House, 1992.

[5] H.R Beom, H.S. Cho, "Mobile robot localization using a single rotating sonar and two passive cylindrical beacons", *Robotica*, Vol. 13, pp. 243-252, 1995.

[6] A. Clérentin, L. Delahoche, E. Brassart, "A multi sensor cooperative approach for the mobile robot localization problem", Proc. of the workshop *Performance metrics for Intelligent Systems*. PERMIS2000, Washington, Aug 00

[7] C. Royère, D. Gruyer, V. Cherfaoui, « Data association with believe theory », 3rd *int. conf. on information fusion FUSION 2000*, Paris, France, 2000

[8] G.A. Shafer, "*A mathematical theory of evidence*", Princeton : university press, 1976.

[9] A. Clérentin, L. Delahoche, E. Brassart "Omnidirectional sensors cooperation for multi-target tracking", *IEEE Int. Conf. on Multisensor Fusion and Integration MFI 2001*, Aug. 01, Germany, pp. 335-340

[10] A. Clérentin, C. Pégard, C. Drocourt "Environment exploration using an active vision sensor", Proc. *IEEE Int. Conf. on Intelligent Robots and Systems* (IROS'99), Korea, Oct. 1999.

[11] D. Gruyer, V. Berge-Cherfaoui, "Matching and decision for vehicle tracking in road situation", Proc. *IEEE Int. Conf. on Intelligent Robots and Systems* (IROS'99), Korea ,Oct. 1999.

[12] Smets Ph., « The nature of the unnormalized beliefs encountered in the transferable belief model », Proceedings of the *8th Conference on Uncertainty in Artificial Intelligence*, San Mateo, Californie, (1992) 292-297

[13] Crowley J., « Navigation for an intelligent mobile robot » - *IEEE Journal on Robotics and Automation*, Vol. RA-1, n°1, pp. 31-41, March 1985

[14] Y. Yagi, Y. Nishizawa, M. Yachida, "Map-based navigation for a mobile robot with omnidirectional image sensor COPIS", *IEEE Trans. on Rob. and Aut.* Vol 11, pp. 634-648, Oct. 1995.

PART I
RESEARCH PAPERS

2M2 - Performance of Robots in Hazardous Domains

1. Derived Performance Metrics and Measurements Compared to Field Experience for the Packbot
   T. Frost, iRobot
   C. Norman, iRobot
   S. Pratt, iRobot
   B. Yamauchi, iRobot
   B. McBride, South West Research Institute
   G. Peri, South West Research Institute

2. Intelligent Robots for Use in Hazardous DOE Environments
   D. Bruemmer, Idaho National Engineering and Environmental Laboratory
   J. Marble, Idaho National Engineering and Environmental Laboratory
   D. Dudenhoeffer, Idaho National Engineering and Environmental Laboratory
   M. Anderson, Idaho National Engineering and Environmental Laboratory
   M. McKay, Idaho National Engineering and Environmental Laboratory

Derived Performance Metrics and Measurements Compared to Field Experience for the PackBot

Tom Frost, Christopher Norman, Scott Pratt, Brian Yamauchi
iRobot Corporation, 22 McGrath Highway, Suite 6, Somerville, MA 02143

Bill McBride, Greg Peri
South West Research Institute, San Antonio, Texas

ABSTRACT

Preparing an Unmanned Ground Vehicle for missions in abusive, dangerous environments requires suitable tests to define the system capabilities. Well-designed performance metrics can provide the government and industry designers with an understanding of how the system should be used in the field and how the system can be improved. This paper describes the metrics and measurements used for testing the PackBot system and compares those metrics and measurements against insights gained in field experience.

I. INTRODUCTION

The PackBot System, shown in Figure 1, is a ruggedized, man-transportable Unmanned Ground Vehicle system that provides a remote presence in dangerous locations. Reconnaissance and manipulation of a remote environment can be performed while the operator remains safe. The PackBot was designed primarily for Mobile Operations in Urban Terrain (MOUT). Designs for situational awareness capabilities and obstacles negotiation capabilities are driven by anticipated urban combat scenarios. The MOUT requirements have resulted in a system that also has many applications in other dangerous combat operations and Urban Search and Rescue (USAR) operations.

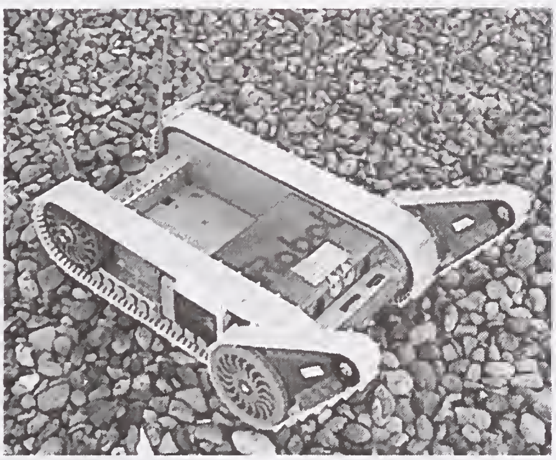


Figure 1. PackBot System

To prepare the PackBot for the hazardous duties that it will encounter in the real world, the PackBot has been tested at iRobot's facility and at the Small Robotic Vehicle Test Bed at the South West Research Institute, SwRI, at San Antonio, Texas. The PackBot system was also exercised at the Army simulated MOUT city at Fort Drum, New York. The predecessor to the PackBot system was tested at a testing ground in Rockville, Maryland. These tests helped to understand and measure the performance of the PackBot before the system was used in the real world. The real world scenario where the PackBot was employed was operations at the World Trade Center (WTC) disaster site. These experiences at the WTC disaster site held many lessons for future Unmanned Ground Vehicle use in the real world. This paper will look at the derived performance metrics and compare these to field experience.

II. PACKBOT HISTORY

The PackBot was developed by iRobot Corporation under DARPA's Tactical Mobile Robotics (TMR) program. iRobot began developing mobile robots for the TMR program in 1997 by creating a proof-of-concept robotic platform designed for MOUT called the Urban Robot. The Urban Robot was designed to be a small man-portable surveillance robot that could negotiate urban terrain. Under subsequent DARPA contracts, the Urban Robot platform became continually more rugged and sophisticated. The later versions incorporated sonar and infrared rangefinders with a more powerful CPU for onboard sensor processing. Under the PackBot contract, iRobot was assigned to develop a more robust and complete robotic system capable of surviving the abuses of real operations. Developing the PackBot became one of the primary focuses of the TMR program and iRobot Corporation was selected as the system integrator.

III. MOBILITY

The PackBot is a tracked robot vehicle designed for use in both urban and wilderness environments. PackBot is equipped with two main treads, used for locomotion, and two articulated flippers with treads that are used to climb over obstacles The PackBot can be fitted with extra treads for additional mobility, see Figure 2. Since the robots developed under DARPA's TMR program were primarily designed for urban environments, this commonality between USAR and MOUT is a large contribution to ease of mission transferability.

Figure 2. PackBot with additional flippers.

Urban environments typically include open spaces such as city streets and building interiors. Common obstacles that robots encounter in urban environments include: curbs, stairs, small rubble piles, pipes, railroad tracks, furniture, and wires. The ability to surmount these obstacles is essential to the success of these platforms.

The PackBot system was capable of traversing all of terrain encountered at the MOUT city at Fort Drum. With very little training, operators were able to drive the PackBot up stairs and through doorways. However, the MOUT city did not have piles of rubble that would be encountered from buildings that have been damaged in explosions or earthquakes.

The SwRI tests consisted of outdoor obstacles. Many of the obstacles, such as pipes and rubble piles, are also informative for predicting performance in urban environments. The obstacle course at the SwRI site consisted of various natural and man-made obstructions selected as a representative subset of robot-scale impediments to cross country movement. The following list is a description of the obstacles used for testing the PackBot and a brief description of the PackBot's performance with each obstacle:

- Railroad ties. No difficulty traversing
- Pipes of ten different diameters ranging from 1.25 inch to 9 inch. The robot had no difficulty traversing these pipes. The robot traversed the largest pipe by lifting itself onto the pipe using its flippers and doing a "backflip".
- Drainage Culvert. 24-inch wide culvert with two 45-degree bends. No difficulty traversing.
- Bamboo Forest. The bamboo forest consists of a matrix of 2 inch PVC pipes on 6 inch centers. The maze width is one pipe-width larger than the width of the robot. Under tele-operation, the PackBot failed to get through the maze on the first try in under the 20 minutes time limit. On the second try the robot was able to pass through the maze in 17 minutes using lessons learned from the first attempt and using the pose capabilities of the PackBot.
- Rock channel. The obstacle has rocks the size of a typical man-packable robotic vehicle. The obstacle was traversed without any problems. The articulators, power, and low center of gravity of the PackBot contributed to the successful negotiation.
- Large, medium, and small rock beds. These beds are populated with rocks that are football sized, softball size, and hockey-puck sized, respectively. The robot was able to traverse all three beds successfully. However, it went out of bounds (off the bed) in one out of the six runs.
- Dirt furrows. The furrows were dry, loose dirt formed into ridges in a 7 ft wide by 30 ft long obstacle. The robot had no difficulty moving through this obstacle.
- Vegetation obstacle. The course was divided into four sections with crops ranging in size from lawn grass to heavy crops, greater than 18 inches high. The light and medium crops were traversed with no difficulty. The heavy crops made the direction of the PackBot difficult to determine and the PackBot moved outside the course in several of the runs. The articulated head/neck unit being developed for

the TMR program will make navigation in thick brush easier.

- Flat Sand Pit. The pit was filled with dry sand and the robot transitioned the pit without difficulty
- Sand Furrows. Dry sand was formed into sand dunes. The robot had no difficulty with this obstacle.
- Mud pit. A pit 4 in by 7 ft by 16 ft was filled with a mud slurry. Figure 3 shows the PackBot after traversing the mud pit. The PackBot could go straight through the mudpit. If the PackBot turned 6-8 times in the center of the pit, the PackBot would become mired. The SwRI report stated "the robot has very good mobility in mud, despite its failure to complete the entire test."

Figure 3. PackBot after Traversing the Mud Pit.

- Inclined ramp. The ramp is adjustable from 0 to 60 degrees. With no payload, the PackBot climbed the ramp, was able to hold position, and was able to skid steer in both directions. With 22.5 pounds of payload, the PackBot was able to ascend and descend up to 55 degrees and able to traverse 45 degrees.
- Curb Height. The PackBot climbed 13 inch curbs.
- Cattle Grating. Metal pipes, 2 inches in diameter, can be moved to different positions. The PackBot traversed this obstacle at all possible gap settings (other than unreasonable ones spaced farther apart than would be expected in an actual animal guard.
- Stairs. The stairs were sets of: wooden 9 inch risers and 11 inch runs, wooden 7.5 inch risers and 11 inch runs, metal 6.5 inch risers and 12 inch runs. The PackBot was able to climb all of the stairs.
- Speed Runs. The speed tests recorded an average (cruising) speed of about 5 mph. With its power booster, the PackBot can achieve burst speeds over 8 mph.

The SwRI tests provided the PackBot designers with excellent information on the PackBot's mobility characteristics tested against precise metrics. Results of the tests were very positive and there was no critical design changes required because of mobility shortcomings.

Since disaster areas and urban terrain are covered in all types of debris, having as much mobility as possible greatly increases the success of USAR missions. Although tracks provide the core of essential mobility, having additional modes of mobility is advantageous. When negotiating rough terrain, robots often flip over. The PackBot's flippers enable it to perform self-righting. The flippers on the PackBot also provide extra mobility, since they are coupled to the main drive tracks, creating a larger adjustable contact surface. The articulated flippers help prevent the robot from being immobilized due to high-centering, enable the robot to climb taller objects, and can help propel the robot forward through dense vegetation through continuous rotation.

At the WTC disaster site, the terrain where USAR operations took place could be divided into two types, the rubble pile, see Figure 4, and buildings for clearance. The rubble pile was not something that these robots had previously encountered and the robots were not specifically designed for the rubble pile terrain. The extreme conditions presented the robots with problems in various areas. The wreckage site had such an incredible amount of debris that mobility was very difficult. The huge pile of twisted steel was challenging for humans to climb, and more difficult still for robots

Figure 4. World Trade Center Rubble Pile.

to negotiate. The mere size of the pieces of steel could be insurmountable. It was very difficult for small mobile robots to traverse such an environment. The PackBot was not able to negotiate the rubble pile except for specific places. For the most part, there were no crevasses in the rubble pile big enough for the PackBot to explore. Usefulness of the PackBot was demonstrated in the surrounding area for building clearance.

The sites requiring clearance and inspection around the rubble pile were strewn with paper and debris. A layer of dust several inches thick covered the area. The PackBot's debris rejection system on the treads was successful, allowing the PackBot to drive through large amounts of debris. This system was previously tested at SwRI with the small rock bed, dirt tracks, and rock channel. The PackBot did not detrack in the WTC operations. The PackBot was used to demonstrate building clearance in the buildings surrounding the WTC. A building clearing operation was performed in an area that had been previously cleared by rescue workers and the PackBot system was shown to be useful for clearing buildings. The PackBot moved through the environment looking for people in the buildings and inspecting structural integrity of staircases. USAR personnel at the WTC observing the demonstration deemed the building clearance capability useful. The PackBot moved through the buildings including many staircases without difficulty except for one staircase. The problematic staircase was covered with dust and the metal was slippery such that the PackBot was not able to get traction on the steps.

The WTC disaster highlighted that specific terrain encountered with USAR operations is difficult to predict, but having a well designed general mobility capability that will climb rubble, self-right, climb staircases, and is maneuverable enough to go through hallways is useful in many scenarios. The designers of the PackBot feel that the terrain such as the rubble pile could not be negotiated without a large increase in complexity and expense in the mobility mechanisms.

IV. DURABILITY

The PackBot was designed from the ground up with considerations for impact resistance, waterproofing, vibration resistance, electromagnetic resistance, low electromagnetic signature, and a wide operation temperature range. The SwRI tests examined the following metrics.

- System Shock. The system survives shocks of 400 G's. 400 G's translates to a 10 ft drop onto concrete. The system was fully operation after multiple drops from 10 ft in positions of rear, forward, and belly down.
- Waterproof. The system is rated for water depths of up to three meters. At SwRI the system was tested in a water channel just covering the PackBot, see Figure 5, and in a pond with depths of up to 5 ft. The PackBot had no problem with the water tests.
- Electromagnetic. The electromagnetic resistance capabilities of the PackBot are restricted from public release.
- Temperature Range. The known operation range of the PackBot is 20 degrees Fahrenheit to 120 degrees Fahrenheit. While the temperature range was not explicitly tested at the SwRI tests, the ambient temperature reached 116 degrees Fahrenheit.

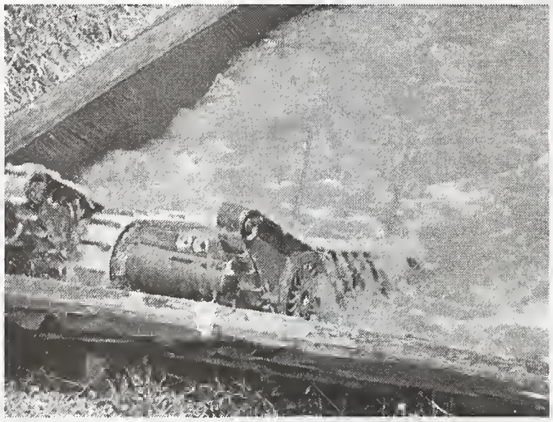


Figure 5. PackBot Emerging From Water Test

The PackBot has an onboard health sensor suite that sends messages back to the Operator Control Unit on the status of the system. Examples of messages sent back to the operator are the thermal sensor readings throughout the system. The system automatically scales back power to motors if the motor is in danger of damage from overheating. The operator can override the thermal protection in the case of a critical mission.

The waterproof capability makes the PackBot more practical for use in the field. The system can be used in rain and can be cleaned with a water hose. The system is also sealed against dust. Many of the combat situations where a robot would be used, such as in Afghanistan, are very dusty environments.

At the WTC, the PackBot's durability contributed to the systems usefulness. The general ruggedness of the PackBot translates to less maintenance and more readiness. The results from the SwRI tests indicated that the PackBot would be a system with excellent durability and throughout the WTC deployment, the PackBot proved to be a durable system.

V. COMMUNICATIONS

The PackBot requires a single 802.11b digital link from the Operator Control Unit to the PackBot vehicle. This link carries the real-time digital video stream as well as the other status information from the robot. In testing, the PackBot's communications link has reached distances greater than one kilometer line of sight. For missions where the system will not be able to maintain line of sight or in an electro-magnetically sensitive environment such as an ordnance deactivation mission, the PackBot system has a fiber-optic spooler that releases fiber optic cable and draws the cable back in. The PackBot also has a payload for RF amplifiers for extended range RF missions. The payload system allows other communication systems to be developed and integrated in a payload slot with ease.

The WTC area, in particular the area directly on the rubble pile, presented significant obstacles for RF communication. The rubble was heavily strewn with attenuating metallic debris and the airwaves were awash with RF radiation from a multitude of radios and equipment. For these reasons, and the fact that tether operations guarantee full frame rate video feedback, the tethered robot systems were used almost exclusively on the WTC rubble pile and were inherently more reliable during operations. Tests that demonstrate a robots ability to pay out its tether and keep the tether from becoming ensnared would be useful. Radio Frequency interference was anticipated as a major and crippling issue once the robots penetrated deep within the rubble.

Exercises were conducted with RF-controlled robots in the area directly surrounding the rubble pile. These areas consisted of blown out, unsafe buildings, wrecked vehicles, and rubble-and-dust-strewn streets. Most of the buildings in this area had not collapsed and, as a result, the areas of robot operation were not as densely congested with metallic debris. This permitted the freedom of RF operation well inside many buildings that were structurally unsound or not yet deemed safe. Although the robots operated freely within many areas of these structures, the operations were conducted with the knowledge that any robots losing communications within unsafe structures were to be considered irretrievable. The dynamic and intermittent nature of RF communications highlights the need for robot-based autonomy to assist the operator during communication blackouts. iRobot is working on autonomous behaviors that detect these situations and react with algorithms designed to navigate or backtrack the robot platform to areas of better communications. Test ranges testing this ability would be of value. In addition, iRobot is working on the ability to use multiple robots relaying communications one-to-another from deep within a RF-impenetrable building. Tests that evaluate this capability would be useful.

VI. SITUATIONAL AWARENESS

Situational awareness, in the context of PackBot missions, refers to the ability of the operator to understand the environment that the PackBot is exploring. This understanding depends on the particular mission and may include:

- Presence of disaster victims, enemy fighters, friendly fighters, civilians.
- Medical condition of detected people.
- Location of Booby traps and mines.
- Integrity of building structure.
- Layout of the area.
- Location of items of interest within the area.

Test courses specifically designed for the above aspects of situational awareness, would be useful.

Selecting the most appropriate sensors for robot situational awareness and understanding how to exploit, merge and interpret the various data provided by the sensors is an extensive area of research. The PackBot can be equipped with various sensors including: cameras, sonars, infrared sensors, and laser scanners. The primary source of information for situational awareness on the PackBot has typically been standard, low-light, or infra-red video cameras.

The PackBot is equipped with a differential GPS system. At the test course in Rockville, autonomous waypoint navigation was demonstrated, see Figure 7. Using only the GPS system, the Packbot demonstrated back-tracking through maze-like courses resembling minefields.

The PackBot is equipped with a compass, roll sensor, tilt sensor, and 3-axis accelerometers. These sensors

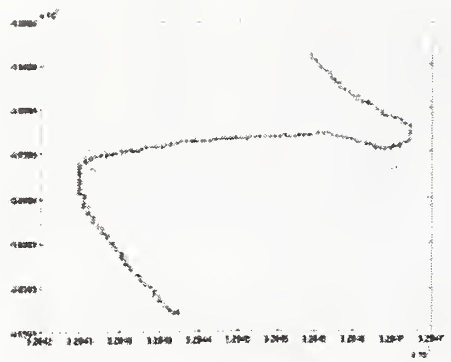


Figure 7. GPS Waypoint Autonomous Navigation

increase situational awareness by providing the operator with additional information on the PackBot's status and position

Many of the tests conducted at SwRI indirectly tested some aspects of situational awareness. For example, the vegetation test, examined the PackBot's ability to determine where it was so that it could move through the vegetation course.

At the WTC disaster, the PackBot used cameras as the main source of data for situational awareness. Both color and b/w video was used. As expected, the color was beneficial for the operator's understanding of the PackBot's environment. The PackBot also used a forward-looking infrared (FLIR) camera. The FLIR was useful in both dark and light environments. Since the environment at the WTC was covered in a thick dust, trying to discern various objects and details of the environment became difficult. The gray dust tended to mute all colors, but having a thermal based view of the world provided an alternate perspective on the immediate environment. Certain aspects of the environment were not readily evident using visible light or low light cameras; however, thermal imaging made these details apparent. The FLIR was particularly useful in detecting the presence and location of people in low-light environments. Tests examining the operator's ability to discern details through a camera would be useful.

VII. DEPLOYMENT

The deployment of a robot system can be broken down into:

- Delivering the robotic platform to the deployment area
- Setting up the control system
- Operating the robot

All of these areas must be well thought out for an Unmanned Ground Vehicle's mission to be successful. The PackBot was developed for ease of deployment. The PackBot is shipped in a padded case that can be lifted by a single person. The Operator Control Unit (OCU) and battery chargers are shipped in a separate case. The robot is deployed by pulling the base chassis out of the case and pushing the flippers onto the chassis. The deployment does not require any tools. The base chassis without batteries weighs 28 pounds so the pieces of the robot can be distributed among a team for carrying to the operation site. Ease of deployment of the PackBot was not tested in a formal setting.

Our experience at the WTC served as an education in the deployment issues associated with real emergency situations. A wide range of scenarios was present at the disaster site, requiring different deployment strategies. Automobiles were used to transport the robot systems and operators from the Javitts Convention Center to the WTC site. At other times automobiles were used to transport the robot systems and operators to a standoff point from which the operators then transported the system to the operation area. Not all areas were easily accessible and some required the PackBot to be carried on foot for long distances (up to approximately one mile). The robot systems needed to be carried over debris and rubble that the robot themselves could not traverse. This highlights the need for robot systems to be lightweight, compact, and man-portable. In other cases, operators carried equipment and robots in their arms and strapped to their backs while riding on ATVs that ferried them to the operation area. The robots did not deliver themselves to the operation area to conserve battery life and because some of the operating areas were not directly accessible.

After arriving at the area of operation, the control system had to be set up. Control stations for the PackBot at the WTC consisted of a human-machine interface of joysticks and buttons for sending commands to the robot. At the time of the WTC effort, the OCU hardware was not rugged or weather-resistant. This resulted in instances where the robot was not used for fear that the system's components were inherently unreliable. Since that time, the OCU software has been ported to more rugged OCU hardware, including a wearable OCU that has been used in military exercises. Having a man-portable OCU was shown to increase the type of missions where the PackBot could be used.

Experience at the WTC illustrated the need to transport the PackBot, OCU, and any other equipment, by foot to the area of operation and the necessity for it to be setup in a short time. A well thought-out container with intuitive locations for each piece of equipment is a necessity. These containers should be capable of being roughly transported in the back of a truck and carried easily. Minimizing the number of cables and plugs is also important. Every piece of equipment and tool needed at the control site must be included in the deployment plan.

During a deployment, the operation of the robot requires an operator's undivided attention. Additional people are needed at the control station to alert the robot operator if the operation area becomes too dangerous. All of the robots were driven under direct tele-operation without any additional autonomous behaviors aiding the operator.

One of the lessons learned at the WTC was the need for a rugged, waterproof robot system that is man-portable and easily set up. As a result, the development of well-packaged rugged OCUs has been mandated for all Tactical Mobile Robots. In order to be effective during these operations and flexible enough to adapt to the varied and extreme conditions associated with USAR missions, the robot systems should require a minimal amount of well-packaged support gear and cannot be limited by the logistics of their support or deployment.

VIII. MODULARITY

The PackBot platform is a versatile platform that can deliver a wide range of payloads that sense and manipulate the environment. Modularity is central to the PackBot design, see Figure 8. Each robot has eight payload interface connectors providing a variety of standard buses and system power. Each standard payload port contains 10/100 full/half duplex Ethernet, FARnet (an iRobot networking protocol), two differential analog video channels, two general purpose digital pins (serial, if needed), USB, and power sourcing or sinking. Payloads that have been designed or are being designed include cameras and lighting units, manipulators, fiber optic spoolers, and hybrid-electric generators. The on-board computer is a 700 MHz, Mobile Pentium III Processor with a 100 MHz system bus and 256 MB of SDRAM. The computing power of the onboard computer is available for running software required for controlling payloads.

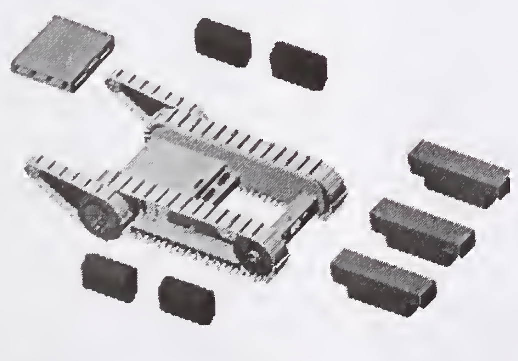


Figure 8. PackBot Modularity.

The PackBot implements a methodology for "snap-on" modular payloads that are quickly and easily interchanged to suit the particular or unique mission at hand. The flexibility and importance of this concept was proven at the WTC site when robots were able to change cameras, lights and tethers as the buildings and areas that were searched presented various technical and physical challenges.

Due to the modular design of iRobot's PackBot, a payload was developed that specifically addressed challenges that were anticipated for the WTC and USAR operations in general. Despite being designed with MOUT operations in mind, the PackBot was adapted to address USAR missions. The payload developed had multiple cameras, infrared illumination, 2-way audio, and a lens cleaning system. In addition to the payloads that were developed in preparation for deployment to the WTC, the standard buses provided on the PackBot payload connectors allowed for several technologies to be incorporated on site. This allowed the robot to make use of sensors and equipment specific to USAR missions. The experiences at the WTC demonstrated the flexibility of a modular architecture and the necessity for robot configuration on a per mission basis.

Payloads used at the WTC were limited to stationary cameras (including a FLIR), light sources (IR and Halogen), up to four battery packs, and Cat 5 cable spoolers. But the versatility, the advantages, and the disadvantages of the overall modular concept were apparent. Even though all payloads were developed for harsh environmental exposure, great care had to be taken when attempting to replace, adjust, or modify payload configurations. Fine concrete dust and debris covered every square inch of the robots' surfaces after every deployment.

Useful metrics for modularity include the amount of time and cost required to add a simple payload to the system.

IX. ENDURANCE

A critical metric for field operations is the maximum length of a mission that can be performed. At the SwRI tests, the PackBot demonstrated run times of 2 hours with constant activity. The maximum mission times can be extended to over 10 hours if the robot is stationary for most of the mission. These mission times represent the standard usage of two batteries. Four batteries can be used to increase the mission time. Endurance tests at SwRI demonstrated that the PackBot can carry payloads more than doubling its 35-pound base weight without significant impact to its mission time. The PackBot achieves its long run times with the selection of power saving components throughout the vehicle.

At the WTC, the lengthier mission times translated to less down time for the robot returning to swap batteries. Although the mission length was not an issue on the specific PackBot missions, there were scenarios at the WTC where a long mission length would be advantages. For example, in conducting extensive building clearance, the operation may be slowed down by the robot returning to the home base every two to three hours for a battery exchange.

X. CONCLUSION

The PackBot performed well at the World Trade Center disaster site in each area of mobility, durability, situational awareness, communications, deployment, modularity, and endurance. Many of the characteristics of the PackBot were well understood from the testing conducted at SwRI, iRobot, and other exercises such as the Fort Drum MOUT city. The application of PackBot technology at the WTC disaster demonstrated that technology developed for MOUT is transferable to and useful in USAR missions although these operations presented unique challenges. The robust mobility and the flexibility provided by the modular nature of the PackBot enabled the robots to operate effectively and adapt to new situations.

The widely varying conditions and environments encountered at the WTC disaster site confirmed that no single size or configuration of robot could address all of the difficulties that an USAR mission can present, however the PackBot provides a useful capability for many situations encountered. The TMR program has made great progress in advancing robot technology and much of what it has done is applicable to areas outside of MOUT missions. The experiences at the WTC show that there is great benefit in using robotic technology in search and rescue operations, yet there remains work to be done in developing specific USAR robot technology.

Additional metrics specifically designed for USAR applications would be helpful. For example, metrics could measure mobility over rubble and building debris, as well as communications range in indoor environments (both intact and damaged). Metrics could also test the ease of deployment by measuring weight and time to setup. The performance metrics developed by SwRI for testing mobile robots provided valuable information about the PackBot's capabilities. The SwRI metrics indicated that the PackBot would perform well in a wide variety of rugged environments, and our experiences at the WTC confirmed the SwRI tests.

Intelligent Robots for Use in Hazardous DOE Environments

David J. Bruemmer, Julie L. Marble, Donald D. Dudenhoeffer,
Matthew O. Anderson, Mark D. McKay
Idaho National Engineering and Environmental Laboratory
Idaho Falls, ID 83415-3779, U.S.A.

ABSTRACT

In *Robotics and Intelligent Machines in the U.S. Department of Energy: A Critical Technology Roadmap*, the DOE has identified the potential for Robots and Intelligent Machines (RIM) to greatly reduce cost, improve worker health and safety, augment product quality and increase overall productivity. In its long-term vision, the DOE has predicted that such RIM capabilities will be as pervasive and indispensable for the DOE and national interests as the PC is today. However, for this vision to be realized, critical issues pertaining to the interaction of humans and intelligent machines must be further explored and new technologies developed. In terms of time, cost and safety, 'usability' may well prove to be the most crucial component of RIM systems for remote handling of radioactive and hazardous materials and a wide variety of other operations. In this paper we examine the metrics used by the DOE to compare baseline radiation survey techniques with a teleoperated robotic survey recently conducted at the Idaho National Engineering and Environmental Laboratory (INEEL). Further, the paper discusses the difficulties and limitations of teleoperation evident from this deployment. To meet the current and future goals of the DOE, it is absolutely necessary to move beyond teleoperation and develop robot intelligence that can be interleaved with human intelligence to mitigate these difficulties. In response to this need, the INEEL has developed a mixed-initiative robotic system which can shift modes of autonomy on the fly, relying on its own intrinsic intelligence to protect itself and the environment as it works with human(s) to accomplish critical tasks.

KEYWORDS: ***Automation, Cognitive science, Human factors, Intelligent robots, Mobile robot dynamics, Robots.***

1. INTRODUCTION

The DOE's Environmental Restoration and Waste Management Robotics Technology Development Program explains that manual work within hazardous environments is slow and expensive. Worker efficiency is low due to protective clothing and, in some cases, exposure limits that require work to be accomplished in several minute intervals. Even when exposure limits are not an issue, fatigue is often induced by confined spaces and by the highly repetitive nature of certain tasks. The cost of a given project is increased because of the special materials needed to protect workers and the environment, and because of the additional wastes generated in the form of contaminated clothing, rags, tools, etc.. Moreover, time required to accomplish missions in hazardous environment is adversely impacted not only by low worker efficiency, but also by the need to prepare the workers and instrument the site.

Consequently, the United States Department of Energy (DOE) continually seeks safer and more cost-effective technologies for use in decontaminating and decommissioning nuclear facilities. As part of the FY 2000 and 2001 Large-Scale Demonstration and Deployment Projects (LSDDP), the Idaho National Engineering and Environmental Laboratory (INEEL) collaborated with the Russian Research and Development Institute of Construction Technology (NIKIMT). This collaboration resulted in the development of the Robotic Gamma Locating and Isotopic Identification Device (RGL&IID) which integrates DOE Robotics Crosscutting (Rbx) technology with NIKIMT Russian gamma locating and isotopic identification technology.

While the new robotic solution offered significant improvements in terms of time, cost, worker exposure and the quality of data acquired, the remote nature of this new technology presented new human-robot interaction challenges. Humans were required to enter the building to instrument the environment with cameras and to assist the robot during the execution of the task. Moreover, during the actual deployment, the robot was only allowed to move at very slow speeds due to the limitations of visual feedback to the operator. In answer to these challenges, the INEEL has developed a dynamic autonomy architecture for the same system used in the RGL&IID deployment.

The new approach presented in this paper permits the robot to take initiative to protect itself and the environment. In fact, the human-robot dynamic has changed from a master-slave relationship to that of a mixed team which allows interaction between peers. When compared with the recent RGL&IID technology, the new mixed-initiative system will remove much of the need for prior instrumentation, remove the need for expert operators, reduce the total number of operators, eliminate the need for human exposure and greatly reduce the time needed for preparation and execution of the task.

2. TELEOPERATED RADIATION SURVEY

Historically at the INEEL, a radiation control technician (RCT) and industrial safety personnel first enter a facility in order to establish accurate conditions for planning purposes. When performing an initial radiation survey, the RCT uses a standard Geiger-Mueller pancake probe to gather radiological information. Once this initial entry has been completed, a video technician may also be required to enter and collect video coverage. Finally a team of sampling technicians is sent into the facility to collect samples used to determine contamination levels and identify which isotopes are present. Typically, this data is then used to aide decontamination and decommissioning (D&D) planning activities (see Figure 1).

Figure 1. Baseline Sample Collection for Laboratory Analysis.

2.1 RGL&IID Deployment

To prove that remote, robotic systems could improve on this baseline, the RGL&IID was deployed in July, 2001 at Test Area North (TAN) 616 (see Figure 2). TAN is located at the north end of the INEEL, about 27 miles northeast of the Central Facilities Area. TAN was established in the 1950s by the U.S. Air Force and Atomic Energy Commission Aircraft Nuclear Propulsion Program to support nuclear-powered aircraft research. Upon termination of this research, the area's facilities were converted to support a variety of other DOE research projects. TAN 616 was built in 1954 as a liquid waste treatment facility. As a result of treating thousands of gallons of liquid nuclear processing waste, there are various levels of contamination present in the facility.

Figure 2. TAN 616.

Three rooms within TAN 616 were surveyed using the RGL&IID: the Operating Pump Room, the Control Room, and the Pump Room. All of these rooms are filled with process piping and equipment at various levels, making make a manual survey very difficult and time consuming to perform. The intent of this demonstration was to gather empirical data to assess the value of using a remote, robotic system. The metrics considered included reduction in cost, accelerated schedule, improvement in safety, and reliability of data.

2.2. Deployment Results

When compared to baseline assessment methods, he most significant benefit of the RGL&IID deployment was the quality of the results relative to the safety of the workers. Although the RGL&IID deployment did not eliminate the need for workers to enter the contaminated area, it did reduce the need for human exposure. The RGL&IID was compared to the following baseline activities: the initial RCT entry, an entry to collect video, and a final entry to collect sample information. The RGL&IID was able to collect dose information, video coverage, and isotopes present in a single unmanned entry.

Radiation exposure to workers supporting the RGL&IID deployment was cut by more than a factor of 10 over baseline activities. During baseline characterization, workers received 82mRem of radiation exposure. During the deployment of the RGL&IID, workers received 7mRem of radiation exposure. In addition, the RGL&IID provided radiation survey results instantly and the complete facility survey was accomplished in 3 days. It took workers using baseline characterization methods 3 months to accomplish the same results. The baseline activities began in August of 2000 and were not complete until November of 2000. Some of the results from the laboratory analyses were not available until January 2001. The laboratory radiological analysis confirmed the presence of Cs-137, Co-60 and Am-241. This same data was available within minutes after the RG&IILD performed the scan.

The deployment of the RGL&IID did require more workers than the baseline characterization. However, during the baseline sampling activities, six entries with as many as six individuals per entry were made, totaling 60 work hours spent in the contaminated area. During the RGL&IID demonstration, only two technicians and one RCT were required to enter the contaminated facility for a total of 10 work-hours spent in a contaminated area. All others associated with the project were able to complete the objectives from outside the contaminated areas. As a result of workers spending less time in the radiation areas, individuals involved in the RGL&IID deployment received 10 times less radiation dose than workers involved in baseline activities.

In addition, the two technicians and one RCT who did enter the facility during the demonstration did so only to assist the movement of the RGL&IID up and down a flight of stairs and to check air quality prior to entering the facility. These individuals maintained as much distance between themselves and the highest contaminated areas as possible. In contrast, the baseline samplers were required to come in direct contact with the contaminated material in order to collect representative samples.

The financial cost of collecting the radiation measurements using the RGL&IID was about half the cost of the baseline technology. In addition to the benefit of significant cost reductions, this technology also generates significantly more data. For example, whereas the baseline survey included 10 point samples, the RGL&IID collected about 20 scans. Each scan covers as little as one square foot or as much as several square feet and may have as many as 64 point measurements. Altogether the RGL&IID deployment resulted in over 200 point measurements that covered over 100 square feet of wall and floor area. The RGL&IID has the capability of providing 100% coverage if needed.

2.3 Limitations to the Teleoperation Approach

Although the 2001 robotic deployment offered a means to reduce human exposure, it did not fully remove the human from the hazardous environment or make it possible for a single human to control the robot. In fact, the baseline survey required only three people, whereas the RGL&IID required six. If robotic systems are to be truly cost-effective and efficient, this ratio of six humans to one robot must be reduced.

Moreover, the data presented above says nothing about the inherent limitations and risks of teleoperation. Teleoperation requires high-fidelity video, reliable, continuous communication, and costly, dangerous efforts to instrument the environment *a priori*. As a mechanical 'subordinate,' the robot was dependent on continuous, low-level input from a human and was poorly equipped to cope with communication failures or changes in operator workload. In fact, while training within a mock-up facility, operators lost control of the vehicle due to a communication failure. Since the last command received by the robot before communications were lost had been a forward acceleration command, the robot continued to accelerate across the room and actually ran right through the walls of an adjacent test bed environment. As a result, the robot's control system was immediately changed to have a "watchdog" system that halted the robot once it recognized that communications had failed.

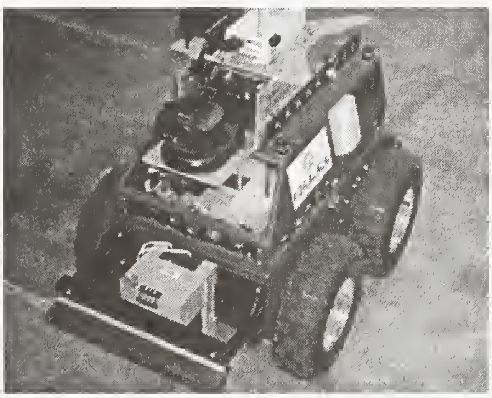
Even so, communication proved to be the limiting factor governing human-robot interaction during the teleoperated deployment. Thick concrete shielding, typical to radiological controls, made it extremely difficult for high-bandwidth communication to support the strictly teleoperated system. As a result, it was necessary for a human to physically place a large antenna directly into the opening of the TAN 616 building. As the robot traveled further from this antenna, the possibility of communication dropouts increased. In fact, operators completely lost contact with the robot at one point during the deployment when the robot traveled out of range. The robot stopped after several seconds once it recognized that communication had been lost. Since the robot was merely a passive tool, it was unable to reorient itself or attempt to reestablish communication. If humans had been unable to enter the environment, the robot would have been lost forever and unable to complete its task. Fortunately, a human was able to move the antenna slightly further into the doorway of the building and communication was reestablished.

The 2001 RGL&IID deployment required weeks of preparation including training operators in mock-up environments. Early on, these training exercises indicated that cameras positioned on the robot would not be sufficient to support teleoperation. The camera could not see the immediate obstacles surrounding the wheels – the very obstacles that posed the greatest threat. As a result, it was necessary to instrument the environment *a priori* with elevated cameras. These cameras were tethered to allow sufficient bandwidth for high resolution video and were set up in the environment by humans. Human placement of tethered cameras is a common practice in nuclear remote inspections throughout the DOE complex. This drawback to teleoperated approaches is further pronounced by the fact that these cameras must be bagged and produce additional contaminated waste once the operation is complete.

Although the cameras were deemed sufficient for the task, operators explained that such a strategy is inherently limiting. The first limitation is that adequate lighting is required to support vision-based teleoperation. Secondly, such cameras are usually unable to provide complete visual coverage. In fact, operators reported blind spots when using the same robotic system and cameras within a different, larger building at the site. In one instance, as the robot rounded a corner and left the visual field of one camera, the last thing the operators saw was the robot begin to tip over. Fortunately, the robot righted itself and was able to complete the task successfully. Nonetheless, the incident emphasizes the need for the robot to provide better feedback and, ideally, to be able to take initiative to protect itself in critical situations.

3.0 MUTUAL INITIATIVE CONTROL

Throughout the DOE complex, teleoperated systems have often failed to address the limitations of telepresence inherent to current communication technologies. On the other hand, attempts to build and use autonomous systems have failed to acknowledge the inevitable boundaries to what the robot can perceive, understand, and decide apart from human input. Both approaches have failed to build upon the strengths of the robot and the human working as a cohesive unit. In response to limitations of both approaches, research efforts at the INEEL have developed a novel robotic system that can leverage its own, intrinsic intelligence to support a spectrum of control levels. We submit that rather than conceive of machines as mere tools or, on the other hand, as totally autonomous entities that act without human intervention, it is more effective to consider the machine as part of a dynamic human-machine team. Within this team, each member is invested with agency – the ability to actively and authoritatively take initiative to accomplish task objectives. Within this schema, each member has equal responsibility for performance of the task, but responsibility and authority for particular task elements shifts to the most appropriate member, be it human or machine. For instance, in a remote situation, the robot is usually in a much better position than the human to react to the local environment, and consequently, the robot may take the leadership role regarding navigation. As leader, the robot can then "veto" dangerous human commands to avoid running into obstacles or tipping itself over.



The resulting robotics system, pictured in Fig. 3., including hardware, software, and interface components, can slide between roles of 'subordinate,' 'equal' and 'leader.' The ability of the robot to change its level of autonomy on the fly supports changing communication, cognitive, perceptual and action capabilities of the user and robot. With the new system, communications dropouts no longer result in the robot stopping dead in its tracks or, worse, continuing rampant until it has recognized that communications have failed. Instead, the robot may simply shift into a fully autonomous mode.

For this system to meet its goals, we must provide robust mechanisms which allow the robot to protect itself and the environment. To do so we fuse a variety of range sensor information including inertial sensors, compass, wheel encoders, laser range finders, computer vision, thermal camera, infrared break beams, tilt sensors, bump sensors, sonar, and others. The robot does not assume that these sensors are working correctly, but rather continuously evaluates its own perceptual capabilities and behavior. Novel sensor-suites and fusion algorithms enhance capabilities for sensing, interpreting, and "understanding" environmental features. Also, a great deal of work has focused on providing situation awareness to the user that can appropriately support the current level of interaction. With the new system we are not limited to visual feedback. Instead, the robot is able to abstract information about the environment at many levels including terse textual descriptions of the robot's local surroundings.

Given the desire to employ robots in hazardous, critical environments, the ability to shift a robot in and out of the leadership role presents a conundrum. The user comes to rely on the self-protective capabilities of the robot and yet, at times, must override them to accomplish a critical mission. For instance, when faced with an unknown boxobstructing the path, the user may shift the robot out of the leadership responsibility for navigation, but grant the robot the "right" to refuse human commands when the physical resistance to motion is beyond a certain threshold. This allows the human to attempt to push the box out of the way without exerting dangerously high force on the robot. For other tasks, the user may need to drive the robot to where it is touching an obstacle in order to take a sample. The user can curtail the robot's collision avoidance initiative and yet customize a "last resort" channel of initiative based on bump sensors and short-range infrared break beams.

Ideally, we need control systems that allow the user to configure the autonomy of the robot on the fly, activating "channels of initiative" that crosscut broad categories. The roles of each team member are bounded by a complex and changing web of capabilities and limitations to which each member must adapt and respond. The ability of the human to develop accurate understanding of robot behavior is essential if this adaptive role switching is to work effectively. One of the most fascinating areas of future work is the need for the robot to be imbued with an ability to understand and predict human behavior.

3.1. Theory of Robot Behavior

The need for human and robot to predict and understand one another's actions presents a daunting challenge. For each level of robot initiative, the user must develop a unique set of expectations regarding how the robot behaves, that is, an understanding or theory of the system's behavior, here after referred to as a theory of robot behavior (TORB). By TORB we mean that the human operator is able to quickly and accurately predict:

1. Actions the robot will take in response to stimuli from the environment and other team members;
2. The outcome of the cumulative set of actions.

In our research we are not concerned with developing a formal model of robot cognition, but rather require that the human understand and predict the emergent actions of the robot, with or without an accurate notion of how intelligent processing gives rise to the resulting behavior. When a human team member is faced with a robot that can orchestrate task elements, the critical issue will not be how the robot or machine "reasons," but rather whether the human team members can accurately predict robotic responses and understand how cumulative actions and responses converge to fulfill task objectives.

Many applications require the human to quickly develop an adequate TORB. One way to make this possible is to leverage the knowledge humans already possess about human behavior and other animate objects, such as pets or even video games, within our daily sphere of influence. For example, projects with humanoids and robot dogs have explored the ways in which modeling emotion in various ways can help (or hinder) the ability of a human to effectively formulate a TORB [1],[2].

Regardless of how it is formed, an effective TORB allows humans to recognize and complement the initiative taken by robots as they operate under different levels of autonomy. The ability to predict and exploit the robot's initiative will build operator proficiency and trust. The development of a theory of robot behavior will also allow the user to switch between and configure the robot's levels of initiative to suit the needs and components of the task at hand.

3.2. Theory of Human Behavior

Just as the human develops a theory of the robot's behavior, the robot must be able to understand and predict the human members of the team in order to adapt to their needs. This is not to say that machines must possess complex mental models or be able to discern our intentions. Rather, it is necessary to raise the level of interaction between the human and robot based upon readily available, non-intrusive workload cues emanating from the operator. The robot's theory of human behavior may be a rule set at a very simple level, or it may be a learned expectation developed through practiced evolutions with its human counterpart. The robot must possess some means to infer the need for intervention. Currently, accurate and non-intrusive collection of these cues is difficult at best, and those measures that have been used are unreliable at worst [3].

The answer to this dilemma is to reduce the human signals down to a prescribed set of channels, which are available as an integral part of the interaction of the human with the machine, and which the machine can use to configure its behavior and level of initiative. Interaction between the robot and human may be through direct communications (verbal, gesture, touch, radio communications link) or indirect observation (physically struggling, erratic behavior, unexpected procedural deviation). Interaction may also be triggered by the observation of environmental factors (rising radiation levels, the approach of additional humans, etc.). The robot's expectations must allow it to recognize human limitations and anticipate human needs without second-guessing the human's every move. When robots do intervene with their human counterparts, the human's TORB must be able to explain why the robot has stepped in and what this shift in control means for the task at hand.

3.3 Dynamic Role Changing

The benefits of allowing the team members to change roles within the team significantly increases team flexibility and reliability in task performance. However, if the interface and human-robot system are not designed in accordance with critical principles of human factors in mind, dynamic role changing may result in mode confusion, loss of operator situation awareness, loss of operator confidence in assuming supervisory control, and degraded and potentially catastrophic performance [4]. Systematic human-centered design is necessary to insure that the robot autonomy conforms to the ways in which humans assign and manage tasks.

Appropriate feedback is required when roles and levels of initiative change. Failure to inform the operator when the robot has overridden commands will lead to distrust of the system, unless the behavior is beneath the level of operator concern. This phenomenon has been studied within the airline industry with pilots and the automatic pilot mode of operation. [5]. Feedback from the robot should not only include the mode change, but also an indication of the reason for the change. For optimal performance of the team, the human must be able to develop expectations regarding when and why the robot will be motivated to initiate a new level of initiative. In order for the human's theory of system behavior to comprehend and exploit robot initiative, the robot's autonomy should be structured hierarchically such that at any given time, the user will know the bounds on what initiative the robot can take. Consequently, the INEEL has developed a control system that supports four clearly distinct levels of human intervention.

4. INTELLIGENT AUTONOMY

Within the last five years, researchers have begun in earnest to examine the possibility for robots to support multiple levels of user intervention. Much of this work has focused on providing the robot with the ability to accept high level verbal, graphical, and gesture-based commands [6], [7], [8]. Others have implemented robots that understand the limitations of their autonomous capabilities and can query the user for appropriate assistance [9], [10]. Goodrich et al. [11] have performed experiments which involve comparing the performance of human-robot pairs using different modes of human intervention. However, very little work has emphasized true peer to peer interactions where the robot is actually able to shift modes of autonomy as well as the user. Sholtz [12] discusses the need for this kind of peer-peer interaction, and provides categories of human intervention including supervisory, peer to peer and mechanical interaction (e.g. teleoperator). Our research to date has developed a control architecture that spans these categories, supporting the following modes of remote intervention:

1. Teleoperation;
2. Safe Mode;
3. Shared Control;
4. Full Autonomy.

For each of these levels of autonomy, perceptual data is fused into a specialized interface (shown in figure 4) that provides the user with abstracted auditory, graphical and textual representations of the environment and task that are appropriate for the current mode. Currently, this interface is used on a touch screen tablet PC made by Fujitsu Corp.. Within this interface, blockages are shown as red ovals and resistance to motion is shown as arcs emanating from the wheels. The robot relays a great deal of synthesized, high-level information (including suggestions and requests for help) to the user in a textual form using the feedback textbox within the image window. Also note that the robot provides textual reports on environmental features at the bottom of the map window and reports on communications status at the bottom of the robot status window. The robot status window provides a variety of information about the status of the robot including pitch and roll, power, heading, speed and a fusion of this information into a single measurement of "health."

The user can move the robot by touching the arrows or may use a joystick or other game controller. It is possible to pan and tilt the camera automatically by touching regions of the visual image. Currently, we are still working to integrate

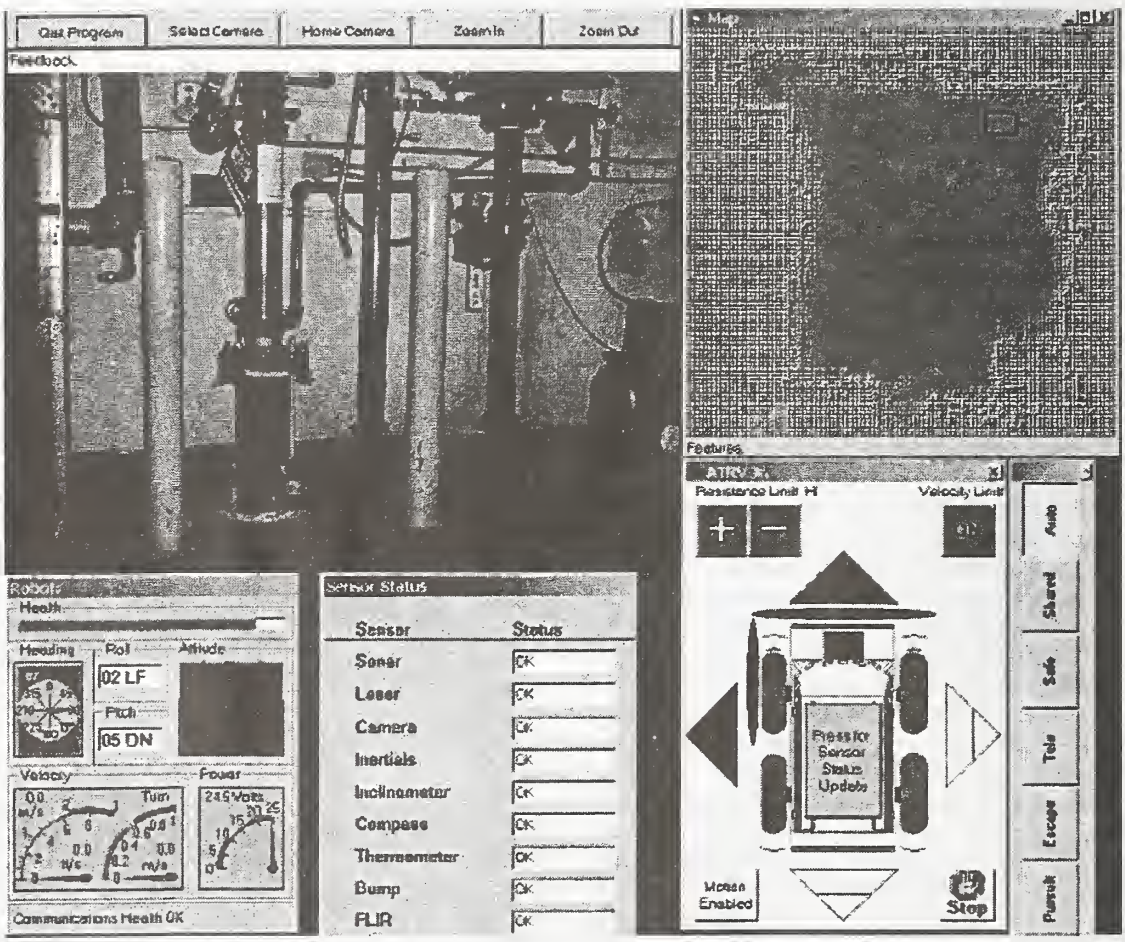


Figure 4: Current interface used for mixed-initiative control of the robot

the on-the-fly mapping capabilities with the interface shown in figure 4. As we continue this task, the interface will allow a number of autonomous tasks (e.g. searching a specified region or going to a goal location) to be issued by interacting with the map itself.

4.1. Teleoperation

We have taken the interaction substrate used in previous INEEL teleoperated robotic systems and revamped it through feedback from people who have deployed such systems. Within teleoperation mode, the user has full, continuous control of the robot at a low level. The robot takes no initiative except to stop once it recognizes that communications have failed.

4.2. Safe Mode

Within safe mode, the user directs the movements of the robot, but the robot takes initiative to protect itself. In doing so, this mode allows the user to issue motion commands with impunity, greatly accelerating the speed and confidence with which the user can accomplish remote tasks. The robot assesses its own status and surrounding environment to decide whether commands are safe. For example, the robot has excellent perception of the environment and will stop its motion just before a collision, placing minimal limits on the user to take the robot's immediate surroundings into account. The robot also continuously assesses the validity of its diverse sensor readings and communication capabilities. The robot will refuse to undertake a task if it does not have the ability (i.e., sufficient power or perceptual resources) to safely accomplish it.

4.3. Shared Control

The robot takes the initiative to choose its own path, responds autonomously to the environment, and works to accomplish local objectives. However, this initiative is primarily reactive rather than deliberative. In terms of navigation, the robot responds only to its local (~ 6-10 meter radius), sensed environment. Although the robot handles the low level navigation and obstacle avoidance, the user supplies intermittent input, often at the robot's request, to guide the robot in general directions. The problem of deciding how and when the robot should ask for help has been a major line of HRI enquiry and will be a major issue in our upcoming human subject experiments.

4.4 Full Autonomy

The robot performs global path planning to select its own routes, requiring no user input except high-level tasking such as "follow that target" or "search this area" specified by drawing a circle around a given area on the map created by the robot. This map is built on the fly and uses frontier-based exploration and localization to perform searches over large areas including multiple rooms and corridors. The user interacts with the map to specify tasks and can guide the robot and infuse knowledge at an abstract level by selecting areas of interest and identifying sensed environmental features, which then become included within the map.

These levels of operator intervention can greatly improve on the opportunities provided to the operators of a strictly teleoperated system such as the one used in the RGL&IID deployment. The human user can switch between these modes to cope with different components of the task. For instance, when a user wishes to move into a new room s/he simply points the robot at a door and then allows the robot to guide itself through the doorway – a task that reportedly took teleoperators many minutes of trial and error.

The latest development, and perhaps the most innovative aspect of our project to date, is that we have imparted a "theory of human behavior" within the robot's intrinsic intelligence, which allows the robot to assess human performance. Before we implemented this theory of human behavior, the robot was already able to use its knowledge of the environment and its own proprioception to take initiative and refuse to accept dangerous commands. However, the level of robot initiative was always controlled by the human. The "theory of human behavior" allows the robot to switch modes when the robot recognizes that the human is performing very poorly. This theory of human behavior is based primarily on the frequency of human input and the number and kind of dangerous commands issued by the user. For instance, if the human has repeatedly placed the robot or the environment in danger, or if the human has been unsuccessful in extricating a robot from a cluttered area, the robot will step in and take over from the operator. Although the human can ultimately override this capability, it provides a means for true peer-peer interaction.

5. PERFORMANCE METRICS & OPERATIONAL IMPACT

The Department of Energy's Robotic and Intelligent Machine (RIM) Initiative has set forth a number of functional objectives to be achieved using robotic and intelligent systems. Some of these metrics include:

- Reduction in exposure to specific hazardous materials;
- Reduction in monitoring costs;
- Secondary waste reduction;
- Productivity increase;
- Production defect reduction. [13]

While these metrics represent increases in "performance" they do not necessarily reflect the full impact of inserting an "intelligent" system into an operation over existing human-centered tasks. Several additional areas that must be considered which are evident within the testing conducted at the INEEL include:

- Operator trust;

- Operator job satisfaction;
- Revised group organization;
- Task skill set adjustment;
- Training re-alignment
- Preparation;
- Consequence of asset loss and contingency / recovery plans.

Consider the reduction of personnel exposure within the INEEL experiment, decreasing from 82mRem to 7mRem and the reduction of total survey time from 3 months to 3 days. These represent dramatic performance gains. Now consider the fact that although remote controlled, the operation of the RGL&IID actually required more personnel, albeit in different roles. The RGL&IID deployment did not eliminate the necessity to utilize a human element for compartment entry, but changed his task from conducting a survey to that of helping the robot ascend and descend a flight of stairs. Also, specific training was required in a mock environment to support the operator's new task of teleoperation.

In addition to the work that the INEEL is doing in remote characterization, some of these same impacts can be seen in the U.S. Air Force and its deployment of the Predator UAV. While Bosnia and Afghanistan have proven the worth of the Predator in remote sensing and ordnance delivery, the Air Force is currently trying to adjust to its operational and organizational impact. Currently being assessed by the Air Force is the proper skill mix for pilots, the correct crew ratio, and training. [14] Additionally, consider the aspect of job satisfaction for a pilot used to flying a plane in the midst of the action now confined to monitoring a Predator control panel miles from the front.

The introduction of an intelligent system be it a mobile or embedded system must be view not only in terms of specific task performance, but also in relation to the overall impact that the system imparts upon how "business" was done in the past. While this change in not necessarily bad or good, it must be examined.

6. REFERENCES

[1] Brooks, R., Brazeal, C., Marjanovic, M., Scassellati, B., & Williamson, M. "The Cog Project: Building a Humanoid Robot." In *Computation for Metaphors, Analogy, and Agents, Springer Lecture Notes in Computer Science, 1562*, C. Nehaniv, Ed., Springer-Verlag, Berlin, 1998.

[2] Brooks, R. A., *Flesh and Machines*, New York: Pantheon Books, 2002, pp. 51-52.

[3] Wickens, C. D., *Engineering Psychology and Human Performance,* (2nd ed.), NY, NY: Harpers Collins Publishers, Inc. 1992.

[4] M. A. Goodrich and E. R. Boer. Designing Human-Centered Automation: Tradeoffs in Collision Avoidance System Design. *IEEE Transactions on Intelligent Transporation Systems*, v.1,no.1, March 2000.

[5] Abbott, K. A., S. M. Slotte, D. K. Stimson, "Federal Aviation Administration Human Factors Team Report on: The Interfaces Between Flightcrews and Modern Flight Deck Systems," Federal Aviation Administration, Washington, DC, Tech. Rep. June, 1996. [Online]. Available: http://www.faa.gov/avr/afs/interfac.pdf, pp. Dl-D3.

[6] D. Perzanowski, A.C. Schultz, W. Adams, M. Skubic, M. Abramson, M. Bugajska, E. Marsh, J. G. Trafton, and D. Brock, Communicating with Teams of Cooperative Robots, In Proceedings from the *2002 NRL Workshop on Multi-Robot Systems*, Washington, D. C., March, 2002.

[7] Kortenkamp, D., Huber , E., and Bonasso, R. P., Recognizing and interpreting gestures on a mobile robot. In Proc. of AAAI 1996. Portland, OR, August, 1996.

[8] Voyles, R., and Khosla , P., Tactile gestures for human-robot interaction. In Proc. of IEEE/RSJ Conf. On Intelligent Robots and Systems, Volume 3, 7-13. 1995.

[9] Fong, T., Thorpe, C., Baur, C., Robot as Partner: Vehicle Teleoperation with Collaborative Control, In Proceedings from the *2002 NRL Workshop on Multi-Robot Systems*, Washington, D. C., March, 2002.

[10] Murphy, R. R., and Rogers, E. 1996. Cooperative assistance for remote robot supervision. *Presence* 5(2): 224-240.

[11] Goodrich, M. A. and Olsen, D. R., Crandall, J. W. and Palmer, T. J. Experiments in Adjustable Autonomy, In *IJCAI 2001 Workshop on Autonomy Delegation and Control*, Seattle WA, August, 2001.

[12] Sholtz, J., Human-Robot Interactions: Creating Synergistic Cyber Forces, In Proceedings from the *2002 NRL Workshop on Multi-Robot Systems*, Washington, D. C., March, 2002.

[13] Department of Energy. "Robotics and Intelligent Machines in the U.S. Department of Energy: A Critical Technology Roadmap", http://www.sandia.gov/isrc/RIMfinal.pdf, October 1998, p.15.

[14] Butler, Amy, Building Predator Fleet Compounds Need for UAV Aircrew Planning, In *Inside the Air Force*, June 21, 2002.

PART I
RESEARCH PAPERS

Plenary Lecture 2 - 2

On communicating with semantic machines

Walter J Freeman

Department of Molecular & Cell Biology
University of California at Berkeley CA 94720-3200
http://sulcus.berkeley.edu

Abstract

Semantics is the essence of human communication. It concerns the manufacture and use of symbols as representations to exchange meanings. Information technology is faced with the problem of using intelligent machines as intermediaries for interpersonal communication. The problem of designing such semantic machines has been intractable because brains and machines work on very different principles. A solution to the problem is to describe how brains create meaning and then express it in information by making a symbol as a representation to another brain in pairwise communication. Understanding of the neurodynamics by which brains create meaning may enable engineers to build devices with which they can communicate pairwise, as they do now with colleagues, though not with words, but with shared actions.

Key Words: EEG patterns, intentionality, neurodynamics, phase transitions, semantics

1. Introduction

The most challenging of the three branches of semiotics is called semantics [5]. It deals with the relation between meanings and representations, a relation often referred to in artificial intelligence and philosophy as 'intention', 'aboutness' (a thought, belief or memory is 'about' something), or 'symbol grounding'. Computers are very good in the other two branches (syntactics, which is the relation of symbol to symbol as found in dictionaries) and pragmatics (the relations between symbol and action like those of traffic signals). They are inept in semantics. The reason for this ineptitude stems from deep differences between brains and computers. von Neumann [19] surmised nearly half a century ago:

> "We have now accumulated sufficient evidence to see that whatever language the central nervous system is using, it is characterized by less logical and arithmetic depth than we are normally used to. ... Thus the outward forms of our mathematics are not absolutely relevant from the point of view of evaluating what the mathematical or logical language truly used by the central nervous system is. [W]hatever the system is, it cannot fail to differ considerably from what we consciously and explicitly consider as mathematics. p. 81"

Brains are exceedingly capable of grasping the salient features of complex situations and social relationships, which are captured in such words as 'value', 'significance', 'import', or 'bottom line', in a word, 'meaning'. It is my conclusion in this essay that meanings exist only in brains, where they take the place of the internal representations that computers use. My conclusion is based on research into the spatiotemporal patterns of active states of brains in animals, that accompany and support the animals' performance of the cognitive tasks involved in learning to respond appropriately to simple stimuli that signify events and circumstances that are vital to their welfare. I find that sensory cortices receive the information that the sensory receptors provide from stimuli, and that this information, once it has arrived in cortex, triggers the construction of activity patterns in brains that constitute the meaning of the stimuli. These patterns over-ride the sensory-driven information [4], which is then discarded, so that everything that an animal learns about its environment has been constructed within its brain from its own experiences.

In order to translate these findings into terms that engineers will require to emulate in hardware the performance of brains in wetware, some further consideration of the biological basis of meaning is required. A meaning state is an activity pattern that occupies the entire available brain [3]. The construction begins with formation within the animal of an activity pattern that embodies its immediate goal, such as food, shelter or a mate, the achievement of which requires acquisition of information from the environment. That information is got by intentional action into the environment, followed by sensory stimulation and learning from consequences of the action. A stimulus such as a light, an odor, or a tone contains information that serves to represent to the animal the state of its environment. It is a material object or process that is equivalent to a book, face, or gesture for humans. It is a part of the environment that has no meaning in itself. The French poet Paul Valéry [18] wrote:

> "I have already explained what I think of literal representation; but one cannot insist enough on this: there is no true meaning of a text. No

author's authority. Whatever he may have wanted to say, he wrote what he wrote. Once published, a text is like an implement that everyone can use as he chooses and according to his means: it is not certain that the maker could use it better than someone else. p. 1597."

My analysis of brain activity patterns shows that sensory cortical activity patterns that are triggered by stimuli are selected by the stimuli but are determined by the history and context of the relations of the individual to the stimuli [4,5]. These brain activity patterns are states of meaning. They occur in the dynamic state space of a brain as trajectories of discrete steps marked by cortical phase transitions. The patterned active states are called wave packets [2]. The way in which they are made by the self-organizing brain dynamics that controls behavior is a pivotal topic in this essay.

The dynamics of brains that creates meaning can be emulated in computer models of brain function [4,12,15]. This step requires that a major problem be addressed: the relation between representation and meaning in brain function. The Shannon-Weaver information theory is representational, because it divorces meaning from information and therefore does not apply directly to brains. Shannon [16] wrote:

> "The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. p. 380."

In Section 2, I sketch some of the principal elements of communication, as a basis for discussing a pathway toward solutions through a better understanding of the biological basis of meanings, which grows from behavioral actions. Meaning stems not from the rule-driven operations between symbols embedded within syntactical systems such as computers, nor from conventional 'computing with words'. it stems from shared actions. In Section 3, I summarize the main observations on sensory cortical wave packets. In Section 4, I enlarge the description to include the limbic system and the origin of intentional behavior. In Section 5, I take up the critical difference between linear and circular causality that underlies the distinction between deterministic and self-organizing systems. In Section 6, I discuss in more detail the relations between meaning and representation. In Section 7, I summarize.

2. Communication by representations

Operational discreteness is essential for communication in dialogue. A pair of brains can act, sense, and construct in alternation with respect to each other, just as dogs circle, and as two humans plan, speak, listen, and hear. Consider brains **A** and **B** interacting [Fig. 1], where **A-B** are parent-child, wife-husband, rabbit-dog, philosopher-biologist, neuroscientist -rabbit, etc. **A** has a thought that constitutes some meaning M(a). In accordance with this meaning **A** acts to shape a bit of matter in the world (a trace of ink on paper, a vibration of air, a set of keystrokes on e-mail, movements of the face, etc.) to create a representation (a sign or symbol for humans, merely a sign for animals, in both cases, information) directed at **B**, R(ab). **B** is impacted by this shaped matter and is induced by thought to create a meaning M(b). So **B** acts to shape a bit of matter in accordance with M(b) in a representation R(ba), which impacts on **A** to induce M(a+1).

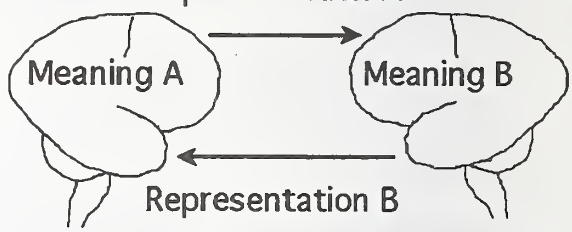


Fig. 1. The schematic shows the roles of representation in the communication of meaning between individuals by the exchange of information through use of representations. A method is proposed to replace one of the communicants with a machine

And so on. Already by this description there is implicit recognition of a discrete recurrent flow of conversation like the tides, so that meanings M(i)'s as constructions of thoughts become the internal active states, and the R(ij)'s as attributes of matter become the external representations. The interchange requires a coordinated succession of phase transitions in both communicants. By its relatively fixed nature an :external" representation can be used over and over, just as we use a letter, word, ideograph or equation. It cannot be said to contain or carry meaning, since the meanings are located uniquely inside **A** and **B** and not between them. The same R induces different meanings M(i) in any other subject **C** who may intercept the transmission of a representation. The objects that are used to communicate are shaped by meanings that are constructed in **A** and **B** iteratively, and they induce the constructions of meaning in **B** and **A** alternately. If communication is successful, then the internal meanings will come transiently into harmony, as manifested by cooperative behavior such as dancing, walking in step, shaking hands, exchanging notes, ringing bells, etc. Symbols persist in books and stone tablets, while minds fluctuate and evolve until they die.

3. Observations on the electric fields, the electroencephalogram (EEG)

A biological approach to the problem of meaning is to study the evolution of minds and brains, on the premise that animals have minds that are prototypic of our own, and that their brains and behaviors tell us what essential properties are common to animal and human minds. Experimental measurements of brain activity (EEG) that follows sensory stimulation of animals show that sensory cortices engage in construction of activity patterns in response to stimuli [2]. The operations are not those that are characteristic of computers, which include filtering, storing, retrieval, template matching, pattern completion, gradient descent, or correlation mechanisms. Each construct is by a phase transition, in which a sensory cortex switches abruptly from one basin of attraction to another, thereby changing one spatial pattern instantly to another like a succession of frames in a cinema.

The transitions in the primary sensory cortices (visual, auditory, somatic and olfactory [1]) are shaped by interactions with the limbic system, which establish multimodal unity, selective attention, and the intentionality of percepts. The interactions of the several sensory cortices and the limbic system lead to goal-directed actions in time and space. Each cortical phase transition involves synaptic changest hroughout the forebrain that constitute learning, so that a unified and global trajectory is formed cumulatively by each brain over its lifetime. Each spatial pattern reflects the content of past and present experience [5], that is, a meaning.

The most important experimental finding is that the neuroactivity patterns in sensory cortex, which form during perception of conditioned stimuli by the animals, are not invariant with respect to unchanging physicochemical stimuli. The brain activity patterns are found to change slightly and cumulatively with any change in the significance of the stimuli, such as by changing the reinforcement, or with the addition of new stimuli [4]. From numerous tests of this kind the conclusion is drawn that brain patterns reflect the value and significance of the stimuli for the animals, not fixed memory traces.

Each pattern forming in response to the presentation of a stimulus is freshly constructed by chaotic dynamics in the sensory cortex, in cooperation with input from the limbic system that implements the supporting processes of attention and intention. It expresses the history, existing state, and intent of the animal rather than the actual incident stimulus. The patterns cannot be representations of meanings of stimuli, either. They are observable manifestations from the material substrate of the meanings that are induced by stimuli, or that emerge from self-induced instabilities in the sensory and limbic systems. Their trajectory constitutes the evolution of a brain in its growth of experience [13]. Similarly, a semantic device must be given opportunity to practice, experience, and grow in abilities to communicate.

The mechanism by which the formation of a wave packet is triggered is of particular interest. When an animal or human receives sensory information, it is carried not by any small number of axons from receptors but by a massive barrage of action potentials. A glimpse of a face, for example, includes all of the detectors for motions, contours, colors, and binocular disparities of the face, and also whatever background against which the face is glimpsed, such as a crowd, a factory or a battlefield. The process in mammals involves a dozen or more specialized areas in each sensory cortex that process the sensory information, with multiple feedback pathways among them.

Despite this enormous complexity, recognition occurs within half a second. The mechanism suggested by EEG analysis is that an entire sensory cortex is destabilized by input that is gated by a rapid eye movement (a microsaccade), or its equivalent in other sensory systems such as a sniff or a finger motion. When it is destabilized, the cortex jumps from one state to another. The transition is completed within 3-7 msec of onset [4, 7]. It is followed within 25-35 msec by the formation of a spatial pattern of amplitude modulation [AM, Fig. 2] of a chaotic carrier wave that persists for 80-100 msec. The AM pattern is accompanied by a spatial pattern of phase modulation [PM, Fig. 2] that is radially symmetric with a fixed phase velocity in all directions. The PM pattern is measured by fitting to it a cone in the 2 dimensions of the brain surface [7].

These two features, the AM and PM patterns, serve to characterize the spatial and temporal location, size, duration and content of the wave packet that is triggered by sensory input. The AM pattern manifests the meaning of the stimulus, not the stimulus in itself, because the AM pattern changes when the context or significance of the stimulus is changed [4] [Fig. 2, right]. In contrast, the location and sign of the apex (maximal lead or lag) of the PM cone are random variables that do not reflect the properties of the stimulus that evoked it or its meaning for the subject [Fig. 3] [1,4,7].

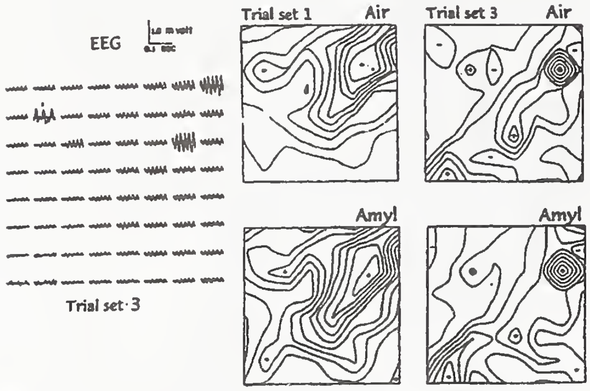


Fig. 2. Left. EEG traces from an 8x8 electrode array (4x4 mm) in 100 msec burst from olfactory bulb after band pass filtering in the gamma range (20-80 Hz). **Middle**. Spatial AM patterns from RMS amplitudes on first trial set with control and odor amyl acetate. **Right**. Two weeks later in session 3 the stimuli were the same, but the AM patterns had evolved to new forms, showing that they are not representations. They are context dependent and individualized for each animal, and they change with variations in the reinforcement.

The randomness of the sign implies that the apex cannot signify a pacemaker for the oscillation, which in any case is aperiodic. The proposed explanation is that the phase gradient manifests the formation of a wave packet by a 1st order phase transition [9], for which the location of the apex reveals the site of nucleation, and the velocity conforms to the finite rate of spread of the state change in a distributed medium, in the case of cortex by the conduction velocities of axons running parallel to the pial surface [7].

The phase gradient shows that the populations of neurons in the wave packet are not oscillating in phase at zero lag, but that they do so with leads or lags that increase with distance from the apex. This feature provides a soft boundary condition for the wave packet, which can be specified by the half-power diameter. The mode and 95% inclusion diameters are shown by circles in Fig. 4, which are superimposed on a diagram of the rabbit forebrain as seen from above. Wave packets having these properties were found in all of the sensory cortices examined. The 8x8 electrode arrays were placed on the primary sensory receiving areas, as shown by the rectangles indicating the size and locations of electrode arrays. The 64 electrodes were used to record the EEGs from the sensory cortices, in order to calculate the AM and PM patterns in the cortical activity.

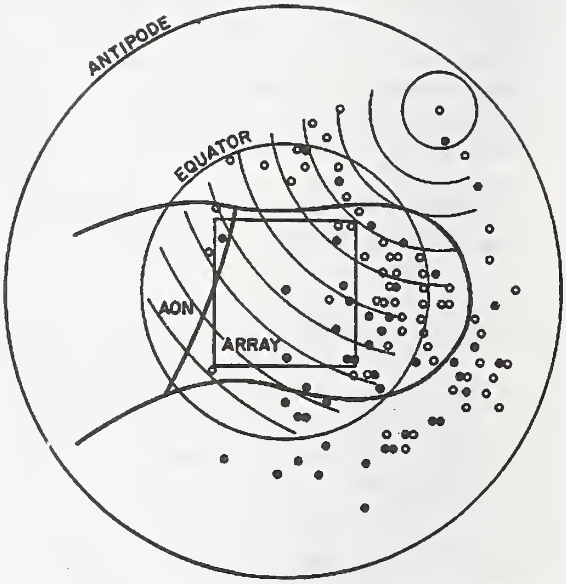


Fig. 3. Phase distributions were measured with respect to the phase of the spatial ensemble average at the surface of the olfactory bulb and fitted with a cone in spherical coordinates. The sketch is a projection of the outline of the bulb as it would appear on looking through the left bulb onto the array on the left lateral surface of the bulb. A representative set of isophase contours is at intervals of 0.25 radians/mm. The locations of the apices of the cones on the surface of the sphere (2.5 mm in radius) are plotted from the center of the array to the antipode. The square outlines the electrode array. The standard error of location of points was twice the radius of the dots. From Freeman and Baird [6].

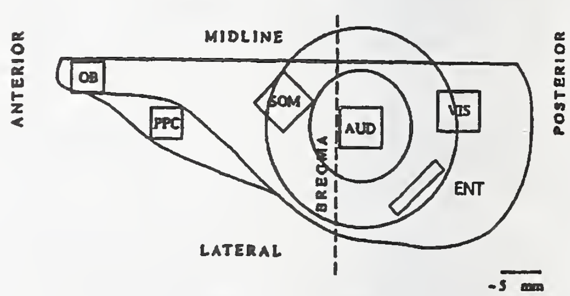


Fig. 4. The outline shows the left cerebral hemisphere of the rabbit as seen from above. The rectangles show the approximate locations of the 8x8 arrays placed on the olfactory bulb (OB), prepyriform cortex (PPC), somatomotor cortex (SOM), auditory cortex (AUD), and visual cortex (VIS), and a 2x8 array on the entorhinal cortex (ENT). The inner circle shows the modal diameter of phase cones. The outer circle shows the diameter including 95% of cases. The vertical line is the zero stereotaxic reference. Diameter inner circle: 15 mm. Adapted from Barrie et al. [1].

The massive size of the areas of cooperation leading to formation of wave packets provides an explanation for the speed with which pattern recognition takes place. The immense cloud of action potentials that is driven by sensory input from a stimulus undergoes a 1st order phase transition that is equivalent to formation of a rain drop from a cloud of water molecules. In the process of the formation of a condensation disk, a spatial pattern of output is selected as the phase transition places the sensory cortex into the basin of an attractor that has been selected by the stimulus. The process repeats at frame rates between 2 and 7 Hz, as shown by the covariance of the successive phase cones with the low frequency oscillations in the theta range of the EEG [7].

4. Neural base of intentional action

The making of a representation is an intentional action. All intentional actions begin with the construction of patterns of neural activity in the limbic system, which has been shown by use of lesions and by comparative neuroanatomy and behavior to be a product of the limbic system [11]. In mammals all sensory input is delivered to the entorhinal cortex, which is the main source of input to the hippocampus, and the main target of hippocampal output [Fig. 5]. Goal-directed action must take place in time and space, and the requisite organ for the orientation is the hippocampus with its 'short term memory' and 'cognitive map'.

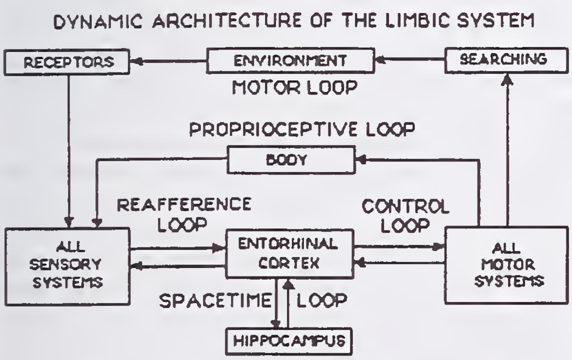


Fig. 5. A schematic diagram summarizes the main elements in the dynamics of intentional behavior to aid in the design of a KIV device capable of intentional action, including semantic communication. From Freeman [6].

For example, hunger is an emergent pattern of neuroactivity that expresses the requirements of brains and bodies for metabolic fuel and building material. It induces a phase transition in the neural populations of the forebrain under the influence of sensory stimuli from the gut and the brain's own chemoreceptors for its chemical state. It is also shaped by neurohormones from nuclei in the brain stem. The emergent pattern impacts the brain stem and spinal cord, leading to stereotypic searching movements that are adapted to the immediately surrounding world. Feedback from the muscles and joints to the somatosensory cortex provides confirmation whether the intended actions are taking place. The impact of the movements of the body on sensory input is conveyed to the visual, auditory and olfactory systems. All of these perceptual constructs, that are triggered by sensory stimuli and are dependent on prior learning, are transmitted to the limbic system, specifically to the entorhinal cortex, where they are combined. When, for example, an animal detects an odor of food, it must hold the immediate memory of the concentration, move, take another sniff, and compare the two concentrations in order to decide which way to move next. The difference in strength has no meaning, unless the animal has a record of where it was when it sensed the first concentration, which way it moved, when the second sample was taken, and to where. This information provides a basis for determining distance and direction in its environment from itself to its intended goal. These basic operations of intentional behavior are properties of the limbic system. The same requirements hold for all distance receptors, so it is understandable that evolution has led to multimodal sensory convergence that performs space-time integration on the multisensory percept, the Gestalt, not on its components prior to their assembly. These operations are already commonplace in robotics, though less so their integration into goal states.

In the description thus far the flow of neural activity is counterclockwise through loops from sensory systems to motor systems, then through proprioceptive and exteroceptive loops outside the brain back to the sensory systems. Within the brain there is a clockwise loop that sustains the flow of activity constituting reafference (the leftward arrows from "motor systems" through "entorhinal cortex" to "sensory" systems, then rightward to close the loops). When a motor act is initiated by the limbic system, it issues a command as an activity pattern descending into the brain stem and spinal cord. Copies of this activity pattern are sent clockwise along these internal pathways to all of the sensory systems by the entorhinal cortex. These 'efference copies' [17] prepare the sensory processors for the impact on the sensory systems of the movements of the eyes, head, ears, and body and, most importantly, the efference copies sensitize the sensory cortices selectively by shaping their attractor landscapes to respond only to stimuli that are appropriate for the goal toward which the action has been directed. The efference copy has also been denoted as a 'sense of effort' [3]. They are the essence of selective attention. These concepts are familiar in feedback control; they need to be generalized in the context of intentional behavior.

5. Linear versus circular causality in self-organizing systems

The conventional view of sensory cortical function holds that stimuli activate receptors, which transmit information to sensory cortex through a linear causal chain, leading eventually to a motor response to the initiating stimulus. Contrariwise, modeling with nonlinear dynamics shows that the stimulus is typically not the initiating event. Rather it is the search for the stimulus that arises in conjunction with an evolving goal in the limbic system, which emerges in a recurrent manner from prior search and its results. This is circular causality at the level of intentional behavior [10].

Much lower in the hierarchy of brain organization is another instance of circular causality. This is the creation of the wave packet in the primary sensory cortex, which consists of the destabilization of a pre-existing mesoscopic state by the introduction of intense barrages of microscopic sensory input. In this case the transition from a prior basin of attraction to a new one, which has been facilitated by limbic modulation, is guided by the sensory input that activates a learned nerve cell assembly comprising a small subset of cortical neurons. The input from the receptors includes both the expected stimulus and the massive receptor discharge evoked by everything that is in the background. The total receptor input forces the instability and triggers the phase transition, and the nerve cell assembly that is activated by the expected stimulus selects the basin of attraction that captures the cortical system. Then the entire domain of the primary sensory cortex transits to the pattern that emerges as the system converges to the attractor in the basin, which in the words of Haken [10] "enslaves" the whole set of cortical neurons by acting as an "order parameter". This new active state has been characterized by Ilya Prigogine [14] as a "dissipative structure", that constitutes, in his words, the "emergence of order out of chaos".

The similarity of the properties of neural activity in the various parts of the limbic system to those in the primary sensory cortices [1,4] indicates that populations of neurons there also maintain global attractors, which are accessed by nonlinear phase transitions, and which are responsible for the genesis of goal states, their motor patterns controlling goal-directed actions, and the reafference patterns that prepare the sensory cortices for the consequences of those actions.

The construction of a device that can simulate the creative dynamics of the brain has been based primarily on the dynamics and architecture of the olfactory system, both in software [8,12] and in hardware [4,15]. The basic unit of the construction is a neuron population called a KO set, that is roughly equivalent to an average neuron [2]. Its time-dependent dynamics is governed by a linear 2nd order ordinary differential equation that is evaluated by fitting its solution as a sum of two exponentials to derive the coefficients for the rate of rise of the impulse response and the passive decay rate of the membranes. Its input is provided by lines that terminate in simulated synapses represented by gain coefficients that are subject to change by learning, and its output is bounded by a static nonlinear gain curve, which is the derivative of the sigmoid curve [6] relating dendritic current amplitude to pulse density output of the population. An interactive population of excitatory neurons is called a KI_e set, and is made by feedback connections between two excitatory KO_e sets in positive excitatory feedback. Similarly a KI_i set is made by feedback connections between two KO_i sets in positive feedback. KI sets have zero and non-zero point attractors and can generate sustained excitatory and inhibitory biases. A KII set is made by negative feedback connections between a KI_e set and a KI_i set. It is has both point and limit cycle attractors.

The interconnection and interaction of three KII sets with distributed feedback delays forms a KIII set, that is capable of aperiodic, nonconvergent, sustained output governed by a chaotic attractor, in addition to outputs governed by point and limit cycle attractors. The nonlinear gain curve, which is the derivative of the sigmoid function, governs pulse density in relation to dendritic current density [2,4,8]. The attractor landscape determines the spatial patterns of EEG of sensory cortices, which are experimentally observed from 8x8 arrays of recording electrodes with a simulation using a KIII set embodying an 8x8 array of coupled KII sets in the OB layer of a KIII set [8]. Each node in Fig. 5 is equivalent to a KIII set. The cooperative synaptic interactions among them support a KIV set, which is responsible for primitive forms of intentional behavior.

6. A hypothesis on the relations of meanings and representations

The idea is proposed that representations are formed by the motor systems through the forward, counterclockwise flow of neural activity. The motor commands are formed at the mesoscopic level by the interactions of neurons and neuronal populations, under the 'enslaving' influence of the global attractor landscape of the KIV set. The commands place the motor systems of the brain stem and spinal cord into appropriate basins of attraction. The representations emerge as spatiotemporal patterns of activity in the effectors of the body (the musculoskeletal apparatus, the autonomic nervous system, and the neuroendocrine nuclei), which respond to the volleys of action potentials from motor neurons at the microscopic level, i.e. single motor units activated in concert. The movements of the body, supported by the autonomic and neuroendocrine back-ups, express the internal meaning states by gestures, vocalizations, shaped objects, etc. These actions change the sensory inflow of the actor in a goal-directed manner. The actions elicit sensory feedback not only to the individual in action. The representations, as intended, activate patterns of receptor discharge in other beings, that place their sensory cortices into the expected basins of attraction. The receivers likewise create patterns of meaning, that lead to up-dates in their limbic systems, re-formation of motor commands, and behaviors that re-transmit representations to the original actor. Thereby, the circular causal chain is maintained between two or more communicators.

The internal clockwise back flow of neural activity serves as an order parameter to modulate and shape the neural activity patterns of the sensory cortices, which transmit the states of their neural populations before and after the expected inputs have occurred, and also if they do not occur as expected, or at all. Modulation comprises not only the reafference but also the exteroceptive and proprioceptive feedback as well. I infer that the organisms constructing and transmitting representations cannot know their meanings until the sensory consequences have been delivered to their own limbic systems. More generally, a poet, painter, or scientist cannot know the meaning of his or her creation until after the act has been registered as an act of the self, nor until the listeners and viewers have responded with reciprocal representations of their own, each with meaning unique to the recipients.

Why do brains work this way? Animals and humans survive and flourish in an infinitely complex world despite having finite brains. Their mode of coping is to construct hypotheses in the form of neural activity patterns and test them by movements into the environment. All that they can know is the hypotheses they have constructed, tested, and either accepted or rejected [5,13]. The same limitation is currently encountered in the failure of machines to function in environments that are not circumscribed and drastically reduced in complexity from the real world. Truly flexible and adaptive intelligence operating in realistic environments cannot flourish without meaning.

This assembly of interacting wave packets may be seen as a mechanism supporting consciousness, which, in the neurodynamic view, is a spatiotemporal pattern of activity that occupies the entire forebrain. It is an internal state variable that has a trajectory composed of a sequence of transitory states that correspond to awareness. Its regulatory role transcends that of the operator in a thermostat, although they share the properties that instantiate the difference between the state of the environment and expectation, such as a sensed temperature and a set point, and that initiates corrective action respectively by intentional action or by turning a heater on or off. The difference is that the simple machine state variable has little history and no capacities for learning or determining its own set point, but the principle is the same: the internal state is a form of energy, an operator, a predictor of the future, and a carrier of information that is available to the system as a whole. The feedback device is a prototype, an evolutionary precursor, not to be confused with awareness, any more than tropism in plants and bacteria is to be confused with intentionality. In animals and humans, the operations and informational contents of this global state variable constitute the experience of causation.

7. Summary

Semantics deals with the relation between meanings and representations, widely known as intention, 'aboutness', or the symbol grounding problem. Brains obtain information about their environments through the consequences of the intentional actions that they execute using their bodies.

Studies of the spatiotemporal patterns of electroencephalographic (EEG) potentials that are induced by conditioned stimuli in the primary sensory and limbic cortices of trained animals have shown that the information thus obtained is used to construct meanings and is then discarded. Computers use representations for information processing and symbol manipulation, but brains have no internal representations. They deploy dynamic neural operators in the form of neural activity patterns that construct and implement meaning but not information. Observers can describe these patterns as information, but that does not imply that the brains do so, or need to. Brains construct external representations of their meanings in the form of shaped objects or movements as their mechanism for expressing their internal states. Examples are facial expressions and gestures in animals and words in humans. Those material constructs are made with the intent to elicit meaning in other brains, but they have no meanings in themselves and do not carry meanings as if they were buckets or placards. Meanings can only exist in brains, because each meaning expresses the entire history and experience of an individual. It is an activity pattern that occupies the entire available brain, constituting a location in the dynamic state space of a brain. EEG data indicate that neural patterns of meanings in each brain are based in mesoscopic wave packets that follow trajectories in discrete steps. Each step is demarcated by a 1st order phase transition that enables formation of spatiotemporal patterns of chaotic oscillations in the gamma range. Amplitude modulation of the carrier wave is the mode of expressing meanings. These wave packets do not represent external objects; they embody and implement the meanings for each individual of his or her interactions with the environment.

Engineers who propose to make semantic machines are faced with the task of defining meaning, which at present exists only in brains, and then with the task of learning how to design machines that can make or cause meaning in themselves. The requirements on network models to simulate the chaotic dynamics of brains include global though sparse connectivity, continuous time dynamics, and distributed spatial functions in two-dimensional arrays of nonlinear integrators. Digital hardware may suffice to emulate the biological functions of sensory cortex in brains by use of nonlinear difference equations as in KIII sets [8], provided that the problems can be solved of attractor crowding and numerical instabilities that inhere in digital representations of chaotic dynamics [12]. Digital simulation is a useful step toward analog simulation in VLSI [15], by means of which to attain the computational speed that will be required for real-time operation of the device. In this way, the next step toward machine meaning can be to use a KIII model of a sensory cortex as an interface between the unconstrained real world, which is infinitely complex, and the finite state automaton that will rely on a dedicated digital computing system as the main support for its artificial intelligence. That is, a model from brain dynamics can provide the eyes and ears for a conventional computer, that can enable the device to interface effectively with the infinitely complex environment that it will share with its designers, and about which it can communicate its views.

Acknowledgment

This work was supported in part by a NASA grant (NCC 2-1244). A preliminary version of this work appeared in Freeman [5].

References

[1] Barrie JM, Freeman WJ, Lenhart M. Modulation by discriminative training of spatial patterns of gamma EEG amplitude and phase in neocortex of rabbits. *J. Neurophysiol.* **76**: 520-539. 1996.

[2] Freeman WJ. *Mass Action in the Nervous System.* New York: Academic, 1975.

[3] Freeman WJ. *How Brains Make Up Their Minds.* New York: Columbia University Press, 2001.

[4] Freeman WJ. *Neurodynamics. An Exploration of Mesoscopic Brain Dynamics.* London UK: Springer-Verlag, 2000.

[5] Freeman WJ. A neurobiological interpretation of semiotics: meaning, representation, and information. *Inform. Sci. 124*: 93-102, 2000.

[6] Freeman WJ, Baird B. Relation of olfactory EEG to behavior: Spatial analysis. *Behav. Neurosci.* **101**:393-408, 1987.

[7] Freeman WJ, Barrie JM. Analysis of spatial patterns of phase in neocortical gamma EEGs in rabbit. *J. Neurophysiol.* **84**: 1266-1278, 2000.

[8] Freeman WJ, Chang H-J, Burke BC, Rose PA, Badler J. Taming chaos: Stabilization of aperiodic attractors by noise. *IEEE Trans. Circuits & Syst.* **44**: 989-996, 1997.

[9] Freeman WJ, Rogers LM. Fine temporal resolution of analytic phase reveals episodic synchronization by state transitions in gamma EEGs. *J. Neurophysiol.*, 87: 937-945, 2002

[10] Haken H. *Synergetics: An Introduction.* Berlin: Springer-Verlag, 1983.

[11] Herrick CJ. *The Brain of the Tiger Salamander.* Chicago IL: University of Chicago Press, 1948.

[12] Kozma R, Freeman WJ. Chaotic resonance: Methods and applications for robust classification of noisy and variable patterns. *Intern. J. Bifurcation & Chaos* **10**: 2307-2322, 2001.

[13] Piaget J. *The Child's Conception of Physical Causality.* New York: Harcourt, Brace, 1930.

[14] Prigogine I. *From Being to Becoming: Time and Complexity in the Physical Sciences.* San Francisco: WH Freeman, 1980.

[15] Principe JC, Tavares VG, Harris JG, Freeman WJ. Design and implementation of a biologically realistic olfactory cortex in analog VLSI. *Proc. IEEE* **89**: 1030-1051, 2001.

[16] Shannon CE. A mathematical theory of communication. *Bell System Tech. J.* **27**:379-623, 1948.

[17] Sperry RW. Neural basis of the spontaneous optokinetic response. *J Comp. Physiol.* **43**: 482-489, 1950.

[18] Valery P. Oeuvres. Etablie et annotee par Jean Hytier. Paris, Gallimard, p1597, 1957.

[19] von Neuman J *The Computer and the Brain.* New Haven CT: Yale University, 1958.

PART I
RESEARCH PAPERS

2P1 - Modeling Intelligence

Modeling Interactive Intelligences

Luis O. Arata
Department of Fine Arts, Languages, and Philosophy
Quinnipiac University
Hamden, CT 06518

ABSTRACT

This paper explores modeling the adaptive mechanisms of autonomous agents. The focus is on reflexive interaction as a looping action that creates a "self."

Imitation and play are adaptive components of reflexive interactions that can provide gradual modifications of the agent's self. Little by little, the autonomous agent reinforces what seems to work and phases out other options through an internal editing process.

A performance metrics for reflexive interaction would have to depend on the task at hand within a context, rather than try to be an absolute measure. Evaluation of performance has to be flexible enough to account for multiple intelligences especially when innovation is possible.

KEYWORDS: *reflexive interaction, open loops, modeling, play.*

INTRODUCTION

This paper explores modeling the adaptive mechanisms of autonomous agents. The focus is on reflexive interaction, which is the making of open loops that combine feedback and feedforward capabilities to interact dynamically with the environment.

Feedback and feedforward can be seen as reaction and proaction, as sensing and probing. An agent that can loop effectively the work of its sensors, probes, and tools would then have the capacity to interact reflexively with its environment. The agent's mechanism of reflexive interaction constitutes a "self."

The agent's looping action creates internal maps to translate interactions and compare them continuously to previous ones stored in its memory in order to adapt to changes. Imitation and play are components of reflexive interactions that can provide gradual modifications to the internal maps of the self in adaptive systems. The mapping process itself can be compared to the writing and editing of a text, where grammar, data, and ideas interact to form an effective map. Little by little, the autonomous agent reinforces what seems to work and phases out other options through an internal editing process.

To explore reflexive interaction, I begin with a presentation of how Rodolfo Llinás describes mapping, and Gerald Edelman's related concepts of reentry and binding. Next, I rely on Jean Piaget's model of adaptation in order to examine the function of imitation and play in an autonomous agent. These concepts come together in the model of the self that Rodolfo Llinás developed as a device that situates the agent in an environment and helps it navigate safely. Finally, I look at mobility of a reflexive agent fueled by two factors: an external changing environment and internal changes that motivate the agent to drift into what Stuart Kauffman calls the adjacent possible.

A performance metrics for reflexive interaction would have to depend on the task at hand within a context, rather than try to be an absolute measure. Evaluation of performance has to be flexible enough to account for multiple intelligences. It should not limit the freedom to develop diverse approaches in the making of agents as well as in the way agents carry out tasks especially when innovation is possible.

1. REFLEXIVE INTERACTIONS

When an autonomous agent performs tasks in an environment, change can happen to both the environment and the agent. This process of mutual change is an interaction [1]. Interaction, rather than action upon objects, complicates the dynamics of a task but it does help model more closely how events happen in the real world, so to speak.

Next, we note that interactions form a link that can create a circulation or a loop between the linked parts. This is reflexivity. The performance of an autonomous agent is affected by its ability to reflect as it interacts.

Two questions then come up. How does reflexivity work to enhance the agent's ability to perform a task? And how could this reflexivity be gauged so that it may be fine tuned with respect to tasks?

In *I of the Vortex*, Rodolfo Llinás develops a fascinating model of the self, based on interactive feedback and feedforward loops. He begins with a view of the brain as a system that does isomorphic sensory-motor transformations of the outside world. This creates representations that help the body act in the outside world.

Llinás then follows the lessons of the sea squirt. This tiny sea creature has mobile state followed by a plant-like one. During the first phase, it has a brain. But when it finally attaches itself to a surface, the sea squirt digests its own brain along with the tail that provided motility. Llinás concludes that "the evolutionary development of a nervous system is an exclusive property of actively moving creatures" [2]. The nervous system and particularly the brain are predictive instruments that allow the organism to move more safely in search of food, often in a potentially hostile environment. The brain creates working models of the environment to give the body interactive navigational capabilities. Llinás imagines that such models are very much dreams of our brain, and in the waking state those dreams are guided and shaped by the senses: "the fact is that we are basically dreaming machines that construct virtual models of the real world." In effect, what we perceive is a virtual world.

The sense of self emerges from interactions in the brain as it coordinates actions. The self is an avatar of sorts within the brain's representation of the world. Our actions follow the displacements of the avatar in the brain's maps of the environment mediated by the senses. In actual dreams, when the senses are dormant, the self moves through a recreated world made of collages of memories patched through internal logics.

Sensations, including the elusive self-awareness, are what reflexive loops feel like in order to help us navigate. Pain and pleasure are guiding sensations. Self-awareness is perhaps the most complex manifestation of this cybernetic system.

In *A Universe of Consciousness*, Edelman describes a reflexive mechanism at work in our own consciousness. It is a signaling process that takes place along reciprocal connections. He calls it "reentry." Edelman sees reentry as the key mechanism that binds all our cognitive mechanisms into a cohesive self. He considers this massively parallel function to be the uniquely distinguishing feature of higher brains. But rather than use reentry as a feature that differentiates higher from lower brains, whatever that could be, we can assume that reentry is to varying degrees a feature of any brain. This feature then can help in the more general modeling of reflexive interaction.

Reflexive neural interaction works within the complex topologies of our brain to create the sense of self out of weaving memories. Edelman suggests that memory is creative rather than replicative: "every act of memory is, to some degree, an act of imagination" [3]. Memory for Edelman is a pragmatic process that always remembers in and from the present. It is simply the ability of an agent to repeat or suppress an action. This ability seems to be at the heart of the sense of self.

When we weave together the work of Llinás and Edelman, we get a rich model of the form and function of the self. According to their combined views, then, it is the reflexivity of the self that would allow autonomous agents to carry out tasks.

The question of gauging and fine-tuning reflexivity with respect to tasks is more complicated conceptually. Llinás speculates that our sense of self and what could be called "intelligence" may well be an emergent property of how our brain wired itself as a navigational tool. He concludes that there are many possible architectures for cognition. Ours does not have to be the only one. In this case, the evaluation of intelligence may have to be done with respect to each separate architecture. In other words, there are multiple intelligences. In this case, Howard Gardner has shown that we cannot have a single measure for all of them. He believes that the notion of assessment has to be reinvented and suggests using simulations to gauge how individuals perform in more realistic and diversified situations rather than using standardized metrics for all intelligences [4]. This implies that for gauging the performance of autonomous agents with respect to realistic tasks, simulations rather than metrics should be used.

2. ADAPTATION, SELECTION, IMITATION, AND PLAY

Piaget presented adaptive behavior as a combination of accommodation and assimilation. Pure accommodation is imitation [5]. Pure assimilation is play. In accommodation, the individual seeks to copy a situation that calls for adaptation and changes following the rules of the external situation. In assimilation, the individual plays with the situation and tries changes it in order to embrace it. In other words, in imitation the individual tends to change the most in the process of copying, representing, or following external imperatives. In play, the person jiggles the external situation and changes it in order to absorb it. Piaget indicated that adaptation involves varying combination of those two extremes.

A key virtue of Piaget's model is that it incorporates naturally the function of play. Imitation has received plenty of attention, but play tends to be underestimated or ignored altogether. Yet it is a key element in agent autonomy, interaction, and development.

A question then comes up. How do imitation and play help stimulate reflexivity in agents?

Play involves a reconfiguration of elements being assimilated. This means that there is a certain metamorphosis at work with the play elements. They are rearranged until something happens. This interaction produces a new meaningful weaving, a new order, a variation, or what could even be seen from the vantage point of a previous order as imperfection or error within the new configuration. But this imperfection becomes innovation when seen from the reconfigured perspective. Such is the creative work of play.

It is important to recognize at this point that adaptation or selection in nature do not yield an exclusive match between the selected agent and the environment. Adaptation does not produce a fittest agent. Edelman has noted that selectional or adaptive systems share a remarkable property: they can use many structurally different ways to achieve similar results. He gave this property the unfortunate name of "degeneracy." We can call it diversification. In evolutionary terms, nature seems to play out all possibilities that yield viable results. Nature tends to favor multiple adaptive solutions. Play is what helps us try out all possibilities. Play is an engine of diversification. Its presence in Piaget's model of adaptation favors the use of multiple points of view or different approaches in the construction of autonomous agents for a given task.

Returning to the relation between play and reflexive interaction, we see that play with its tendency for overflowing boundaries, testing constraints, and diversifying, tends to excite the agent so that it has to constantly readapt. This exercises reflexivity.

But play can make or break an agent. Play needs to be bounded somehow so that it allows the agent to exercise its reflexivity without pushing it past a breaking point.

Imitation, on the other hand, is linked with representation. This helps the agent map its environment in connection with given tasks. We need to distinguish, however, between copying structures imitated, and transforming them into maps based on the agent's system of representation. Imitation for autonomous agents is then a transformation and a translation from something perceived outside of the agent to something inside the agent that allows it to interact more effectively with what is perceived. Imitation makes maps that are webs of memories. The sense of memory is the one we saw before based on Edelman's view. It is non-representational. Woven memories, although not a copy of what is perceived, do evoke it in a functional way. The agent uses the map of memories to help with navigation, placing itself in it.

We can take a closer look at the form and function of play in a created autonomous agent. How would it work? Could play have a purpose? First of all, it is important to recognize that play is interactive. It does not rest entirely on the side of the player. The player needs a partner. That partner is outside the player. This is perhaps the single most reason why play has been mostly overlooked before in its cognitive function. It has not been noticed that there is play in the environment. Natural environments give us room to play. They invite play. That may be why children play. As adults we tend to play less because we need to function in created structures that are often set in their ways and restrict play. These created structures lack the flexibility of natural environments. Our constructed environments do not allow for play, unless they are playgrounds or have been designated as toys. Our non-artistic creations come with built-in purposes. Deviations from expected uses are usually not welcomed.

How could play enter into autonomous agent design? What enhancement of the agent could it bring about? To think about play in relation to created autonomous agents we need to have an uncertain environment to begin with. If everything in the environment is determined, if rules of operation are fixed, if goals are absolute, then there is no room to play. But if rather than goals we think of tasks, if the journey is at least as important as the destination, and if the environment has the potential for surprises, then we can think about play.

Surprise is the order of the day in laboratories, for example. It is unfortunate that theorist shun Murphy's laws. Can a created autonomous agent also play when given a task in an uncertain environment? How can we design it with that type of intelligence? How do we gauge the ludic capabilities of an agent? The introduction of flexibility into designs is a first step. It is a passive response to play. The next step is to design agents that can assimilate as they play. I think we don't even have preliminary solutions modeled after these questions because play has hardly been a factor in design. But we can make some observations.

First of all, the agent has to be able to alter rules. Secondly, the agent has to have tools that can be used in unintended ways. Let's consider now Piaget's sense of play as assimilation of the environment to the individual's existing structures. For an agent, those structures have to be open, flexible, so that they can interact in unexpected ways. Secondly, the agent's tasks have to be defined in fuzzy terms so that there is room to play. Perhaps finally, operating rules can be allowed to yield new tentative combinations that could yield unexpected results. The original rules should not be discarded as new ones emerge. They all go into a widening repertory of behaviors and models.

We can say then that the agent learns through play about its own system and about the environment. Assessment of the ludic side of an agent could then be linked to the quality of what it learns with respect to very broadly specified tasks. Contrary to what is often said, play does have a non-trivial function. In a created autonomous agent this function could be the performance of self-motivated activities in an environment that invites tinkering and exploration. This generates discovery and learning new ways.

The value of play is well understood in the arts. An actor plays a role because there is room for interpretation and self-expression that can yield surprise and improve the performance as gauged by audience response. Salvador Dalí used to say that to innovate one must first master previous techniques. When we play with techniques and tools using pre-existing knowledge, then something new can emerge within that set of elements. Picasso liked to point out that painting wins in the end—not the painter. This underscores that the agent at play cannot have full control of the actions. The agent opens up and exposes itself to the environment to invite the unknown in and play with it. The writer Annie Dillard perhaps summed this best by observing that the art object "is a cognitive instrument which presents to us, in a stilled and enduring context, a model of previously unarticulated or unavailable relationships among ideas and materials" [6].

Perhaps we can use more effectively as a model for created agents Llinás' conclusion that we are dreams guided by the senses. The construction of the agent's self has to incorporate internal reflexivity. It has to allow somehow for self-creation using Piaget's sense of assimilation. Play allows created agents to have autonomy in uncertain environments. The internal structure of the agent has to be able to learn from such play and place in memory what it considers valuable. An agent's cognitive structure can be designed so that it recognizes new objects by playing with them to detect actions associated with the object and turning such associations into usable knowledge. Play, then, is a feedforward interactive behavior: it tests and tags new objects through tinkering. It has a quick trial-and-error component that can probe the environment and see what fits the agent's tasks and behaviors. Play can be seen as a form of communication with the uncertain and the unknown.

3. DISPLACEMENTS

Finally, I would like to touch upon issues of agent displacement as they relate to play. What self-motivates autonomous agents move or change? Conversely, what would prevent autonomous agents from drifting away from preset tasks? Since we are focusing on autonomy, we can exclude direct external influences such as instructions given periodically to the agent, or built-in engines.

We can imagine that two factors may affect autonomous agents. One is passive and due to the change of the external environment because this would tend to affect the agent's functioning. The other is internal change that comes about as the agent interacts in new ways with the environment. This second factor may be active if it becomes entangled with the agent's sense of self and the agent feels in control of changes.

Stuart Kauffman proposed in *Investigations* that biospheres are constantly reorganizing and innovating [7]. He noted that within this uncertain environment, autonomous agents have the tendency to propagate their systems of organization into adjacent possibilities, and create new order. Of course, such displacements and interactions would affect to some degree the agent's internal systems. The broad implication is that nature may be constructing itself through the interactions of autonomous agents. But Kauffman is not suggesting that an evolutionary vector is at work here. This process of creation is a natural drift, as Francisco Varela once proposed [8]. We can see it as the result of play from the part of agents.

Nature seems to go for viability rather than optimization. This may well be because in a complex environment it is simply impossible to optimize, particularly when there is interaction between guests and host. The way to proceed is to diversify viable options in a given environment and let them evolve. This is where the function of play becomes critical: it stimulates diversification. From a design perspective, we can call this multiple modeling. This agrees with what Llinás indicated for cognitive systems: they can have many possible architectures.

The difference between evolution and drift may well be mostly a matter of perspective. As Varela noted, Darwinian evolution favors optimization, whereas natural drift calls only for viability. But viability becomes optimization when selection criteria become so stringent that there is only one option left in the end. For practical purposes, it is better to require the less stringent test of viability. This gives the agent more room to play. Optimization needs a clear definition of a landscape in the first place, which in a natural environment is a daunting if not impossible task. Viability does not. It is self-testing, so to speak.

This suggests that performance evaluations of autonomous agents in a natural environment or any other environment subject to unpredictable changes should be based on viability rather than optimization. One possibility is to gaûge the quality of play by the number of viable solutions that an agent can produce for a given task in an environment. Control or enhancement of the agent's displacements may affect its performance. Running models to tweak their parameters may help gauge such displacements and fine-tune them for specific tasks. Play control would require building boundaries that focus and restrict interactions, as well as insulate the agent from external changes. Play enhancements would come about by opening boundaries to give the agent more freedom in certain chosen directions.

Play then fuels reflexive interactions between the agent and its environment, as well as between the agent's self and its maps. These mechanisms can help the autonomous agent adapt to an environment to carry out tasks that yield viable behaviors and outcomes.

4. REFERENCES

[1] Arata, Luis O. "Reflections about Interactivity," 1999. Internet location: http://media-in-transition.mit.edu/articles/index arata.html.

[2] Llinás, Rodolfo. *I of the Vortex: From Neurons to Self.* Cambridge: MIT Press, 2001. P. 17.

[3] Edelman, Gerald M. and Giulio Tononi. *A Universe of Consciousness: How Matter Becomes Imagination.* New York: Basic Books, 2000. P. 101.

[4] Gardner, Howard. *Intelligences Reframed: Multiple Intelligences for the 21st Century.* New York: Basic Books, 1999. P. 207-10.

[5] Piaget, Jean. *Play, Dreams, and Imitation in Childhood.* New York: Norton, 1962.

[6] Dillard, Annie. *Living by Fiction.* New York: Harper & Row, 1982. P. 184.

[7] Kauffman, Stuart. *Investigations.* New York: Oxford, 2000.

[8] Varela, Francisco, Evan Thompson, and Eleanor Rosch. *The Embodied Mind.* Cambridge: MIT Press, 1991.

STRUCTURED APPROACH TO THE INTELLIGENT SYSTEM DESIGN

Dr. Leonid M. Polyakov
Professor of the Department of Computer Systems and Math
Globe Institute of Technology
New York, NY
Leonid@globeinstitute.org

ABSTRACT

Artificial Intelligence (AI) is a science of intelligence system design. Existing definitions of intelligence don't answer some important questions of engineering procedures. What kinds of intellectual tasks do we have? Who is more intelligent or smarter: a scientist or a wood-maker (human or machine), a metal-maker or a wood-maker? How to design a system with reasoning as the most powerful intellectual function? What is intuition? Can we design a system with intuition?. All these topics are subjects of discussion in this paper. The goal of this paper is to find active, productive may be not the best way to determine the starting position and some directions of intelligent system design.

Keywords: *intelligence, intuition, associative thinking, fuzzy, agent classes, intelligence classes, structure, reasoning, preposition logic, predicate logic, knowledge base, rules of reasoning, application rules, design.*

INTELLIGENCE DEFINITION

There are many different definition of intelligence [1--19], but none of them give the answer acceptable by a scientific community.

First of all, intelligence is a fuzzy term. In some cases it is very difficult to draw a line between intelligent and non-intelligent natural and artificial systems. For example, biological adaptation or any kind of evolution can be presented as learning intelligent ability or non-intelligent process. It is difficult to determine when expert system became an AI system. All intellectual activities are triggered by the goal. "A system can be intelligent only in relation to a defined goal…" [11]. All kinds of intellectual activities in the specific area are based on knowledge, but intelligence is not knowledge. Knowledge is a "tool" of intelligence. If you don't understand a goal, you are not capable to reach it. An ability to learn is an important intellectual ability that can improve knowledge. Knowledge reinforces intellectual activities. There are two components of intelligence: general intelligence that is inherited at birth, and knowledge-based intelligence that can be improved by learning. Twin studies support this approach but the twin result of intelligent level measurement depends on intelligent definition and the measurement method that still are problems. Professor Ulric Neisser (Cornell University) notes [30] that in isolated areas where the gene pool has been unaffected by migration, the longer that children attend a school, the higher their I.Q.'s on average. The knowledge base is a module, organized memory of an intelligent system and knowledge is just a content of this base. "Intelligence is an internal property of the system, not a behavior" [20], but a behavior is the main criterion of an intelligence level. This level can be determined by a test. The natural system inherits strong information through genetic code. They have very strong general intelligence. The artificial system has relatively weak information power from the hardware and the software.

Inherited "brain power" of natural intelligence is determined by power of a neuron net (number of neurons and power of connections: value of a weight function, a threshold and a transfer function). A process of knowledge collection creates an information flow through the neuron net and increases power of connections (Hebb). As a result "brain power" increases. In a simple brain model a neuron is a variable with two values: ON and OFF. The simple rule of knowledge (if…then) in KB can be presented as a variable with two values. The more rules the more connections between the variables the higher intellectual power of a system. In AI systems that are not based on neuron net technology, increasing a number of rules in KB increase a number of virtual connection between the different parameters as well. The knowledge base is the main source of information and intellectual power of artificial systems. Inheritance is the main source of natural intelligence power.

A definition is not a description of a system design. Good definition presents a term from the user (customer, supervisor, etc.) point of view and helps to recognize it among the other terms. It should be as simple as possible.

Now we can try to design an intelligence definition. **influence mental and physical behavior in accordance General intelligence (inherited or hardware intelligence) is an organized combination of conscious and unconscious potentials (cognitive and expressive potentials) in a sentient system that able to direct and with a system goal.** General intelligence is a capability opposite to ability of the system. It can be evaluated indirectly through electrical and chemical brain activities are measured by instrumentation. A level of fuzziness determines a level of confidence.

Knowledge-based intelligence can be defined as **a** ***knowledge-based*** **general intelligence (or ability) of a** ***domain-oriented*** **system to act under** ***existing constraints*** **(limitations) and reach external or internal** ***goals*** **or** ***decrease the distance*** **between the starting and the goal's stages. A goal's description can be presented in crisp, fuzzy, or probability and statistics theory languages.** This definition covers not just cognitive power but a power of sensing system and the actuators. In this case cognitive power is limited by knowledge and extended by learning. Knowledge-based intelligence can be evaluated by behavior tests. General intelligence of AI systems can be evaluated by reading of design documentation and program source code. Unfortunately access to this information usually is not available under secrecy conditions. Both of these definitions of intelligence agree with existing two-factors, multiple-intelligence, and information-processing theories of natural intelligence [23]. This is the extreme definition. As a working definition of AI system it is possible to accept: **the system with one or more intellectual abilities (Fig 1) or the system that emulate one or more intellectual abilities.**

Note: a condition statement "if-then" in a hard coded program is not an element of a knowledge base. The conventional closed-loop information system (control system) is not knowledge-based. Only the intelligent system is based on knowledge. This statement supports fuzzy nature of intelligence definition.

The time it takes to execute the goal is one of many important characteristics of a system performance such as learning ability, duration of the object recognition, etc. and should not be incorporated into the definition. The statement "...a goal should be reached for a certain period of time..." does not make any sense and does not make the definition better.

In some discussions we can hear that sometimes a high intelligent system performs some specific job worse then a lower level of a intelligence system (in human society we have the same). So what, don't use the tractor instead of the hammer. The right choice is a very important characteristic of human and artificial intellect as well.

CLASSIFICATION OF THE INTELLIGENCE TASKS AND ABILITY OF THE AGENTS TO ACHIEVE THEIR GOALS

The system design is based on set of desirable system tasks (abilities) and relationships between them. A conventional software design technology creates the programs for the specific problem solution. From the programmer point of view AI is a software design technology to create programs with intellectual abilities. These programs can be used for wide area of the problem solutions.

What kinds of intellectual tasks do we have? Who is more intelligent or smarter: a scientist or a wood-maker (human or machine), a metal-maker or a wood-maker? In [23] we can read: "Who's more intelligent: a Supreme Court Justice or professional golfer?" Task classification can help to design system.

Intelligence abilities can be presented as the multilevel structure [2,32]. But this structure presents a system view from one side. A **multilevel structure of functions (abilities)** (see Fig. 1) with expressive and cognitive thinking at the upper levels of the structure; learning, problem solving, and etc at the middle level; and generalization, reasoning, conceptualization, induction, information collection, perception, etc. at the lower level of the structure presents the system from another point of view. Perception can be presented as a set of the different signal, emotion – as a set of the different kinds of emotions. Conceptualization itself consists of two levels: identification of important characteristics and identification of how the characteristics are logically linked. Certainly this structure is based on some level of simplification of the relationship as well as the set size of abilities. But any way, this structure can help to determine the set of abilities related to the certain goal, their relationships, and determine the metric structure to evaluate the system intelligence levels. It takes longer to exercise the upper level abilities than the lower level abilities. Different tasks need different sets of abilities to fulfil these tasks. "Animal behavior ought to be used as a model to define a hierarchy of intelligence tasks"[28].

The structure of the intelligent functions was discussed early, for example, in [20]. In accordance with the definition in this paper "intelligence is an ability..." but what kind of abilities are "the information and values the system has stored in its memory"? The mixture of different levels like reasoning and problem solving (reasoning is the lower level ability relatively to problem solving), reward and punishment with value judgement (reward and punishment is the lower level ability relatively to value judgement) creates the wrong structure. Computation power (speed, sophistication of the algorithm of computation or something else?) and number of processors, knowledge representation mechanisms and symbols (symbols of what?), and many others are placed in one row as dimensions of intelligence.

In [1] and [2] we have a very clear answer to the goal importance problem. The goal is a result of the intelligent system actions. "A system can be intelligent only in relation to a defined goal or environment"[11]. Different tasks, different areas of activities have different goals. Similar goals can be combined into one class, which we can call the goal class. The **goal class** (similarity) is determined by minimal set of abilities to fulfil the goal of the task with the same weight functions of each ability. All agents that exercises the same minimal set of abilities to carry out the goal with the same set of weight functions can be combined into one class which we can call the **agent class.** The members of the same **agent class** can fulfil the goals of the same **goal class**.

A scientist, a wood-maker, and a metal-maker are trained to perform different classes of tasks (goal classes) and we cannot make any comparisons between different agents of different agent classes. So, it is impossible to compare a scientist and a handyman, as long as they fulfill different tasks under different goals. In some cases it is possible to combine the systems with visible different intelligence levels into one agent class. For example, agent from the "handyman class" and agent from the "scientist class" can be combined into one class if these systems act under **similar** goals as for example, surviving, reproduction, repairing something that does not need any special scientific knowledge, etc. Performance of these systems and level of their intelligence can be compared. Multiple-intelligence theory [23] supports this point of view. Achievement of the same goal by the different agents usually involves the same set of their abilities with the same set of weight functions. It is impossible to compare a car and bookstore even if you use the money scale to evaluate them. But as soon as you look at them as investment choices (taxi or shop), you will be able make a comparison: the same goal (profit) and the same set of characteristics. The stock market permits the use money scale to compare almost everything because the same investment goal and the same parameters of evaluation. Good gamblers in reality use vector function, but non-sophisticated people play by price difference.

It is reasonable to suppose that a scientist has better training in abstract abilities than a handyman. It is reasonable to make serious decision about differences of the intelligence level of these systems. Different domain applications are determined by different sets of abilities. But it is possible that a handyman (human or machine) has grater level of intelligence (special abilities) then a scientist (human or machine). If these handyman's special extra abilities are not fit to the his/her/its kinds of activities then they can not be utilized in the professional activities of a scientist and a handyman as well. Performance of the different tasks utilizes the certain limited sets of intelligent abilities. In this case a very smart metal-maker will not be able to use full his/her/its available intelligence power and will not be able to demonstrate the full set of abilities that are not important to fulfil standard metal-worker task.
In order to make an evaluation of a real "brain" power of the system, we should assign a reasonable and comparable goal level. It is important to avoid using the overqualified agent. By the way, it is a big problem of the job market.

Human intelligence is not a subject-oriented set of abilities. We are not talking about a genius; we are talking about ordinary people. I myself don't understand the nature of genius. Machine intelligence (for the time being) is a subject-oriented ability. There are different levels (capacities) of intelligence. Sometimes different levels of performance (skills) can be presented as different levels of intelligence. Different levels of performance are determined in many cases by limitation of one or more elements of the system. Advanced upper level abilities of the intelligent structure (generalization, conceptualization, etc.) are not guarantying a high level of the skills. For example, low capability of the sonar sensors can prevent a person to be a musician even if he/she/it has a suitable capability of the rest of the subsystems. Beethoven was not a deaf man; he lost his ability to hear. Composer as music designer can "hear" his music with his inner "sensor". The famous woman Helen Keller, author and educator was deaf, blind and mute but she had a sensitive tactile system and sense of smell. She learns to "hear" and to speak and she was able to make her great intellectual power work [14]. A scientist with a high level of intelligence may have a problem doing a manual job if he/she/it does not have suitable actuators. A "handyman" is not a handyman without hands. There are two choices to design the definition of intelligence: to extend definition and include sensors and actuators or to add separate explanation of sensors and actuator importance. As soon as we talk about intelligence as "...an ability of a system to act appropriately..."[1], we include an actuator into this definition. No sensors – no knowledge, no actuators – no performance; and it is impossible to evaluate the level of intelligence.

Globe Institute of Technology has strong positive experiences to reeducate people of different backgrounds into very good programmers. Our experience shows that a medical doctor, a psychologist, an engineer, a teacher, and people who worked in many other fields, can fulfill tasks of the high level programmer and they like doing it. There are a lot of people who are good scientists, medical doctors or have other professions and at the same time are good writers or musicians, biologists or good mathematicians, etc. These examples support the assumption that human intelligence is determined by a goal achievement activity level but not by area of application. In other words, agents in many different domains can be combined in one agent class if they exercise the same minimal set of abilities at the same levels.

AGENT WITH REASONING. THE STRUCTURE DESIGN

Reasoning is the most powerful intellectual function but it is not easy to emulate it. The main problem is determined by the nature of reasoning that is based on computation with words instead of computation with numbers. There are a lot of different approaches to the knowledge representation in the agents. The most important languages of knowledge representation are preposition logic and predicate logic. Agent models of reasoning based on preposition and predicate logic are topics of this discussion.

Reasoning, as we know, is the process of drawing conclusion from facts. There is a lot of research dedicated to the problems of reasoning and the agent structure design [7,9,18]. All of them are based on representation of knowledge as rule-based, semantic net, or frame structure knowledge base. These knowledge bases (KB) include just application knowledge (AKB) (domain oriented KB). Rules of reasoning are applied on AKB in different ways for different agents. This approach decreases the level of universality of the agent. Most existing systems with reasoning are not universal theorem provers (http://www-formal.stanford.edu/clt/ARS/Entries/acl2). These systems are based on rules of reasoning and don't work with application knowledge. Some of them, like ACL2, are designed as multi-KB with (Deductive machinery, Dynamics, Persistence). However, all these systems are based just on preposition logic. The most interesting result in the area of reasoning is the Jess language (Jess, the Java Expert System Shell http://herzberg.ca.sandia.gov/jess/demo.html). This language is based on just one KB-AKB. Information is presented by predicate logic. Rules of reasoning are incorporated into a source code.

A possible way to increase the level of universality of the agent is by creating the double KB agent structure. The first KB is application knowledge base (AKB); the second one is rule of reasoning KB-RKB. RKB is universal KB. It can be used with different AKB. The Double-KB structure of a system (the programmer Mr. U.Rozenblad) is shown on Fig.2. Complicated application rules should be decomposed to simple rules by DeMorgan's, associative, and other laws. The idea of a multi-KB in search engines also was described by Dr. Lotfi Zadeh in "The Prototype-Centered Approach to Adding Deduction Capability to Search Engines- The Concept of Protoform" (BISC letter, 21 Dec 2001) http://www.cs.berkeley.edu/People/Faculty/Homepages/zadeh.html . In this letter: "The deduction database is assumed to consist of logical database and a computational database, with the rules of deduction..." Rules of deduction are Implication Elimination, And–Elimination, And-Introduction, etc. These rules transfer rules of application in canonical form. Transformation can be done during of application role presentation or during a program execution. First way is more time efficient, second one does not change of application roles presentation and make them easy recognizable.

Advantages of reasoning rules separation from a program:

1. Simple choice of the set of rules from the prepared list of rules for each area of application.
2. Standardization of a program by coding only reading functions and functions of recognition. The standard program can be easy designed and testing. The standard program has a high level of universality and can be easy adapted to the different areas of application.
3. Separation of rules of reasoning from a program makes a program easy readable, better understandable, and as a result more reliable.

INTUITION

It is not the question: does machine have intuition or doesn't have it? If we are machines and we have intuition then a machine has intuition. The problem is to define the word intuition to make it worktable. There is a lot of different definitions [3,8,10,12,15,16,18,21,24,25,31,32,34]. From the practical point of view we need the positive, constructive approach even if in the beginning we design system just with the realization of the simple process of the intuition imitation.

The most famous definition of intuition is "the immediate knowing, or learning of something without the conscious use of reasoning; instantaneous apprehension" (Webster's New universal unabridged dictionary). The difference between intuition and association (by Webster's) is: the first is a non-conscious process, the second is a conscious process. This definition is not productive.

It is impossible to extract knowledge from nothing. If you never heard about the stock market or brain surgery, you will never have intuitive decision in these areas. Knowledge extraction is a conscious process. There are two conditions under which one idea is able to recall another. "These conditions may be classified under two general heads, the *law of contiguity* (in reality is **law of associations**), and the *law of associations* (in reality is **law of reasoning**). The first states the fact that actions, sensations, emotions, and ideas, which have **occurred together, or in close succession, tend to suggest each other** when any one of them is afterward presented to the mind. The second indicates, or ideas tend to recall their like from among previous experiences. On their physical side the principles of association correspond with the physiological facts of **reexcitation of the same nervous centers**" (Webster's New universal unabridged dictionary). These two definitions relate to two different processes. One is associated thinking, second one is intuition.

The memory is a network hierarchy [23]. It is arrangement of nodes or categories such that concrete ideas are at the bottom of the hierarchy and are connected to more abstract ideas above them. The most abstract ideas are at the top. **Intuition is the process of searching a problem solution and ideas along the hierarchy of a memory. Association is the process of searching a problem solution and ideas through direct relationship between them.** Analogy is based on semantic similarity, similarity of memorization time combination of the objects in the set based on different criterions, etc. Intuition is a result of free "travel" through the memory structure. The typical example of associative learning is a baby learns to associate the smell of its mother

with food (classical conditioning). A student learns that working hard usually produces good grades (operating conditioning) [26]. This definition is worktable, reasonable and non-contradictable. Such presentation of intuition may not be the best but is very productive for artificial intelligence system design. In [33] there is description of personality with intuition as a personality with "focus on **implication** and **inferences**."
Associative thinking creates the net between different objects, events, and images. Intuition is not just the search of the similar solution of the problem but sometimes is "design" of a solution as a sophisticated assembly of the several elements. In this case we deal with more complicated procedures. From the external point of view intuition looks like associative thinking.

In opposite, the research and decision searching are ***motivated*** intendment organized processes of a solution of a problem searching. Importance of intentionality is mentioned by many philosophers. For Edmund Husserl (German philosopher) intentionality is "*one* essential feature of any consciousness". For Jean-Paul Sartre (French philosopher and writer) "intentionality is consciousness".

Spontaneous brain activities can be triggered by a non-verbal fuzzy defined problem that is dominated in the memory at this particular time. In this case, accidental knowledge activates the algorithm searching for patterns, history, relationships and etc. to find solution of the problem. The more data and information that is stored in the memory, the better the result of the intuition process. The higher information diversity the more efficient an intuition solution. There are two kinds of information: genetic and non-genetic. In artificial systems genetic information is stored in the hardware and partly in software and contributes to the artificial intuition.

Intuition is an "automated" high speed process. Cognitive thinking in most cases is a low speed "manual" process executed under control of human will.

All knowledge about objects and processes has to be presented as models designed from the different points of view (structural models, math models, logical models, chemical models, electrical and information model, etc). For example, a human body can be presented in the different ways as a structured model, a chemical model, an information model, a mechanical model, etc. Such ways of knowledge presentation make it possible to easily identify common features in different areas. The structured organization of the knowledge in the memory is a very important condition of effective functioning of the artificial intuition. In our case, (artificial system). we don't have problem of the natural brain in attaching meaning to the symbolic representation [21]. Existence of the memory makes reasonable the materialist point of view and cognitivist point of view as well [21]. Anyway this problem is not a subject of our discussion.

In the reconstruction of new knowledge when any past event or experience is recalled, the act of recollection tends to bring again into use other events and experiences that have become related to this event in one or more of certain specific ways this association. Associative memory refers to the ability to recall complete situations from partial information These systems correlate input data with information stored in memory Information can be recalled from even incomplete input. Associative memory can detect similarities between new input and stored patterns [10, 12, and 21]. So intuition and association should work together. Realization of the associative memory can be done as the Hopfield Neuron Network [21].

Spontaneous brain activities can be triggered by spontaneous interest of the system to the problem. For example the problem of dangerous environment for the system existence. It can be the cause of spontaneous problem formulation. Spontaneous undependable problem formulation is possible just in case of the availability of the powerful sensor system. This system collects information about simple, separately non-dangerous events, puts it together independently from a human will, looking for patterns and creates a sense of danger. The process and information are presented in fuzzy description.

Let us look at the simple scenario. At nighttime you left a party with your friends and were going home. You were thinking about the good time you had. Suddenly you step-into a dark street as a part of your way home (level of darkness may be different-fuzzy descripted). Nothing is wrong around but your body becomes alerted even if you try to calm yourself through reasoning. **Intuition vs. reasoning!** Intuition can win because reasoning is based on the same knowledge! Reasoning can just add some new information and knowledge. As a result, correction of the sense and behavior can be obtained.

When we meet a stranger, we receive a complex of information about his/her appearance, body language, and way of talk, etc. Our brain compares this information with the fuzzy or statistical models of a "good" or "bad" object appearance, behavior, etc. and creates our "fuzzy" impression model about this object. "Good" or "bad" object models are based on our previous experience. This situation was emulated on the computer (the programmer Ms. N. Elisseeva). The system was able to generate intuitive impression at the meeting with a stranger (Fig. 3).

Unintentional brain activity can include the testing procedures. One day I sent an e-mail but forget to attach my file I promised to my friend. I was sure that I did not make a mistake. In the middle of the night, I suddenly woke up and realized that I did not attach the file. My brain was testing my activities stored in the short memory against the goal procedure and sent me the error massage. It's remind automatic virus testing software when we rebut the computer without special activation. Certainly, it is just analogy. This ability the control a human activities is a very useful part of Artificial Intelligence. The "Testing" module can do this test.

The described approach can be illustrated by another example. Suppose we have the AI system, which has extensive working experience in the different areas of knowledge and powerful learning ability from the experienced external teacher. The knowledge is represented as the models: linguistic, math, logical, structured, etc, above. Suppose the system has knowledge about a damper, which is presented as a linguistic model (damper, controller of acceleration, brakes, and etc) and as a math equation of a damper:

$$y=(1+e^{-t/T})$$

or the transfer function

$$W(p) = k/(1+Tp),$$

as a physical object (hydrodamper, pneumatic damper, capacitor, inductance, robber damper, spring, mechanical brakes, electrical brakes, flywheels etc), and so on. All these models create the hierarchical structure in the knowledge base. The more abstract the description, the higher location levels. The linguistic description belongs to the higher level. The physical description belongs to the lower level. Suppose we have a control system and would like to reduce the acceleration of the moving parts. The AI system has information (through the sensors) about the problem and starts looking at a solution of the problem without our interference. The search procedure is shown in Fig.4. Each level represents a new level of goals. Each new goal motivates a next search step.

Intuition can be activated in the slipping stage when the brain is working without participation of the human will. In an interview with The New York Times (Nov. 14, 2000) Dr. Terrence J. Sejnowski (a neuroscientist at the Salk Institute in San Diego) said: "There has always been a close connection between sleep and creativity, which may be a byproduct of the way that nature chose to consolidate memories".

CONCLUSION:

1. Intelligence consists of two parts: inherited and developed.
2. Intelligence abilities can be presented as the functional multilevel structure. Similar goals of the agents can be combined into the goal **class.** The goal **class** is determined by minimal set of abilities to fulfil this goal of the task and one set of weight function.
3. All for each alternative – class member agents that exercises the same minimal set of abilities and common set of weight functions to carry out the goal can be combined into the **agent class.** The members of the same **agent class can fulfil** the goals of the same **goal class.**

4. The structure of intelligence system should be designed as a two-knowledge base system. One is an application knowledge base, another one is a reasoning knowledge base.
5. Intuition is the process of searching a problem solution and ideas along the hierarchy of a memory. Association is the process of searching a problem solution and ideas through direct relationship between them.

REFERENCES:

[1] Albus J., Outline for Theory of Intelligence. IEEE Transactions on Systems, Man, and Cybernetic, ol. 21, No 3. May/June, 1991.
[2] Albus James S., Meystel Alexander, Behavior Generation in Intelligent Systems, NIST.
[3] Artificial Intelligence With Dr. John McCARTY Conversation On The Leading Edge Of Knowledge and Discovery With Dr. Jeffry Mishlove, 1998.
[4] Boden, Margaret A., Artificial Intelligence and Natural Man, Basic Books, Inc., New York, NY, 1977.
[5] Bock, Peter, The Emergency of Artificial Intelligence: Learning to Learn, The AI Magazine, Fall, 1985.
[6] Charnik, Eugene and McDermott, Drew, Introduction to Artificial Intelligence, Addison-Wesley Pub. Co., Reading, MA 1985. .
[7] Cawsey A. The Essence of Artificial Intelligence. Prentice Hall, 1995
[8] Computers and The Mind with Howard Rheingold. Conversation On The Leading Edge of Knowledge and Discovery with Dr. Jeffry Mishlove, 1998.
[9] Dean T., Allen J., Aloimonos Y. Artificial Intelligence. Theory and Practice. The Benjamin/ Cummings Publishing Company, 1995.
[10]Decision Support and Expert Systems1997. Management Support Systems by Efraim Turban. Prentice Hall. 1995.
[11] Finkelstain Robert, A Method For Evaluating the "IQ" of Intelligent System, Preliminar Proceedings "Performance Metrics for Intelligent Systems Workshop, August 14-16, 2000, Gaithersburg, MD.
[12] Foundations of Neural Networks by Khanna, Addison-Wesley, 1990.
[13] Huffman Karen, Vernoy Mark, Vernoy Judith, Psychology in Action, John Wiley & Sons, Inc
[14] Keller Helen. The Story of My Life, 1902.
[15] Language And Consciousness. Part 4: Consciousness and Cognition with Dr. Steven Pinker. Conversation On The Leading Edge Of Knowledge and Discovery With Dr. Jeffry Mishlove, 1998.
[16] Meystel A. Evolution of Intelligent Systems Architectures. What Should Be Measured?
Performance Metrics for Intelligent Systems. Workshop. August 14-16, 2000, Gaithersburg, MD

[17] Mind Over Machine With Dr. Hubert Dreyfus. Conversation On The Leading Edge of Knowledge and Discovery with Dr. Jeffry Mishlove, 1998
[18] Mind As A Myth with U. G. Krishnamurti. Conversation On The Leading Edge of Knowledge and Discovery with Dr. Jeffry Mishlove, 1998.
[19] Myers David G. Psychology, Worth Publish, 1995.
[20] Negnevitsky M. Artificial Intelligence. A Guide to Intelligence Systems, Addison-Wesley, 2001
[21] Neural Networks and Physical Systems with Emergent Collective Computation Abilities by Hopfield, J. Proceedings, Natural Academy of Sciences USA 79, 1985. Their Minds, PHOENIX, 1999.
[22] Panos Antsaklis, Defining Intelligent Control. Report of the Task Force on Intelligent Control, IEEE Control Systems, June 1994.
[23] Plotnik R. Introduction to Psychology, Brooks/Cole Publishing Company, 1995
[24] Philosophy, History & Problems by Samuel Enoch Stumpf, McGraw- Hill, 1994.
[25] Psychology by Peter Gray, Worth Publishers, 1999.
[26] Psychology by Crider A. B., Goethals G. R., Kavanaugh R. D., Solomon P. R. Harper Collins College Publishers, 1993.
[27] Russell Stuart, Norvig Peter, Artificial Intelligence. A Modern Approach, Prentice Hall, 1995.
[28] Subhash Kak, Grading Intelligence in Machines: Lessons from Animal Intelligence, Preliminary Proceedings "Performance Metrics for Intelligent Systems Workshop, August 14-16, 2000,
[29] Winston, Patrick Henry, Artificial Intelligence, Addison-Wesley Pub. Co., Reading, MA 1985
[30] The Rising Curve, edited by Proof. Ulric Neisser, American Psychological Association, 1995.
[31] The Transcendence Of The Ego. An Existentialist Theory Of Consciousness by Jean-Paul Sartre. Hill and Wang-New York, 1997.
[32] The Oxford Companion to MIND, edited by Richard L. Gregory, Oxford University Press, 1987
[33] Tieger P.D. and Barron- Tieger B. Do What You Are. Little, Brown and Co. 1995.
[34] Unlocking your Subconscious Wisdom. Part 1: Using Intuition with Dr. Marcia Emery. Conversation On The Leading Edge Of Knowledge and Discovery With Dr. Jeffry Mishlove, 1998.

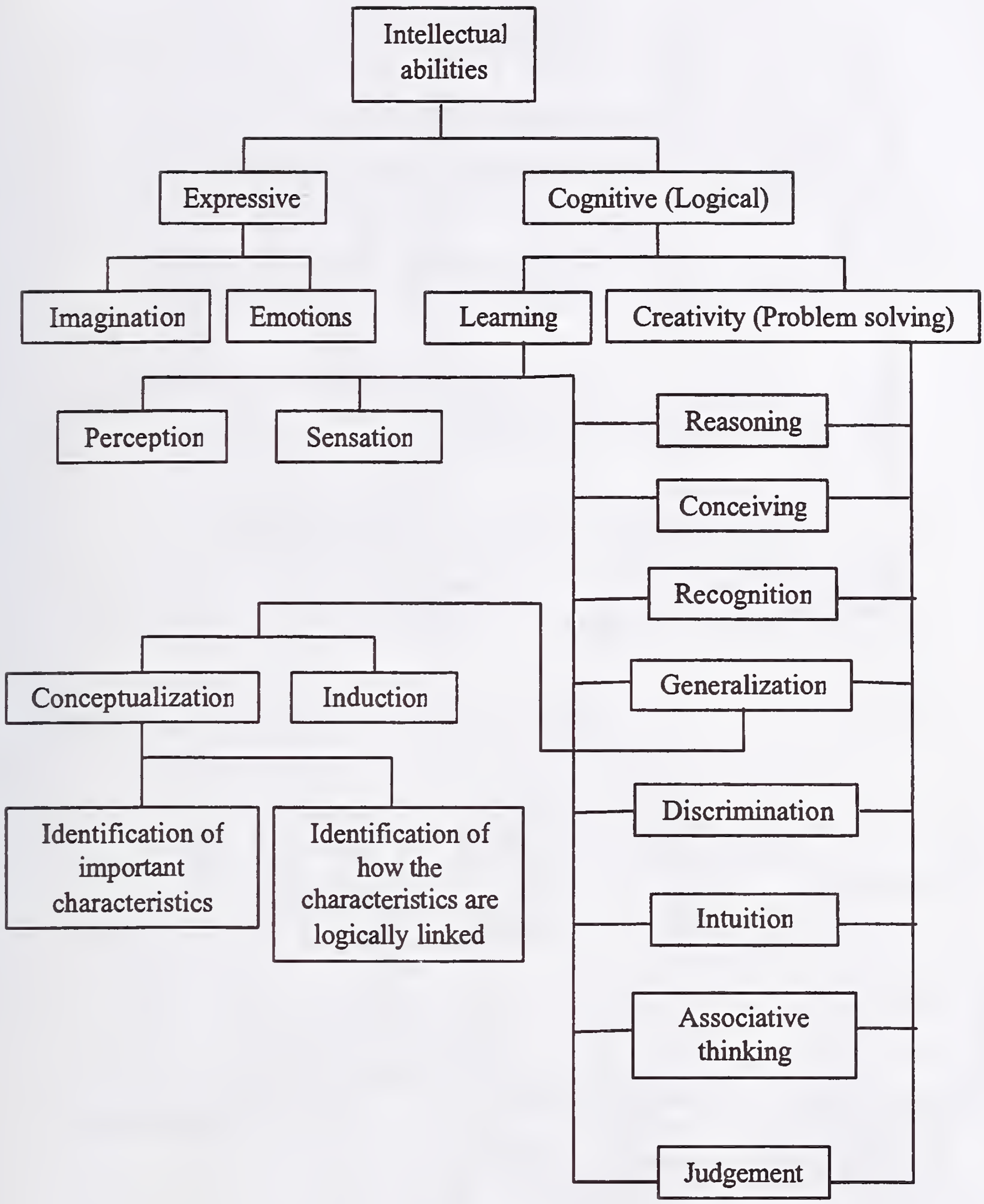


Fig. 1.Structure of the most important intelligent abilities

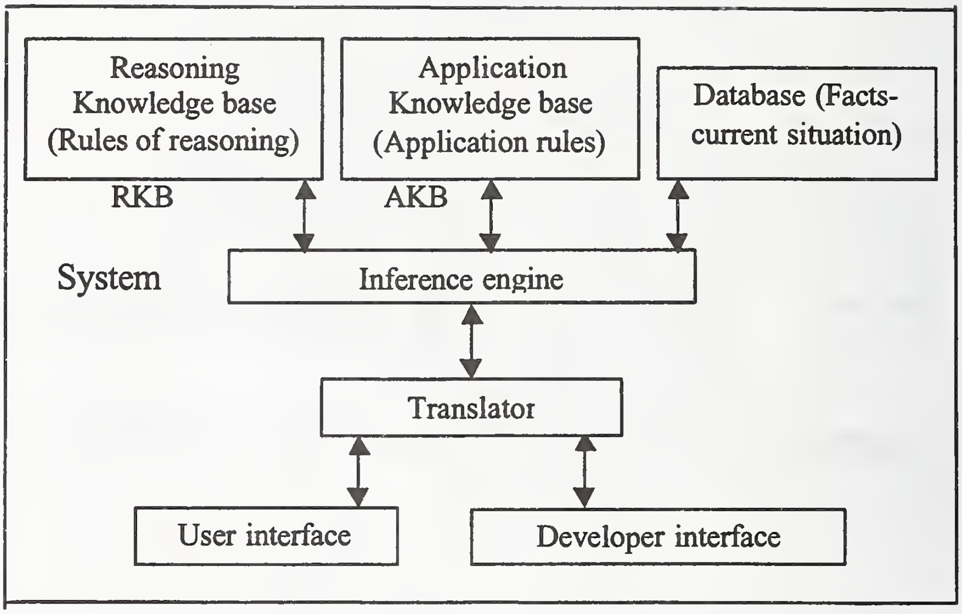
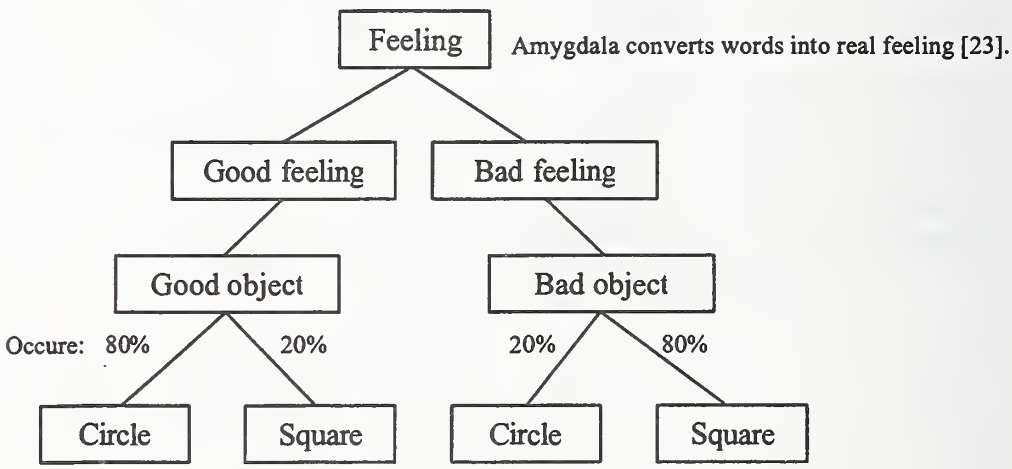


Fig. 2 The double-KB system structure



Level of good feeling from a circle: 0.8 – 0.2 = 0. 6
Level of good feeling from a circle: 0.8 – 0.2 = 0. 6
"Good" and "bad" can be described as fuzzy variables.

Fig. 3 The memory structure.

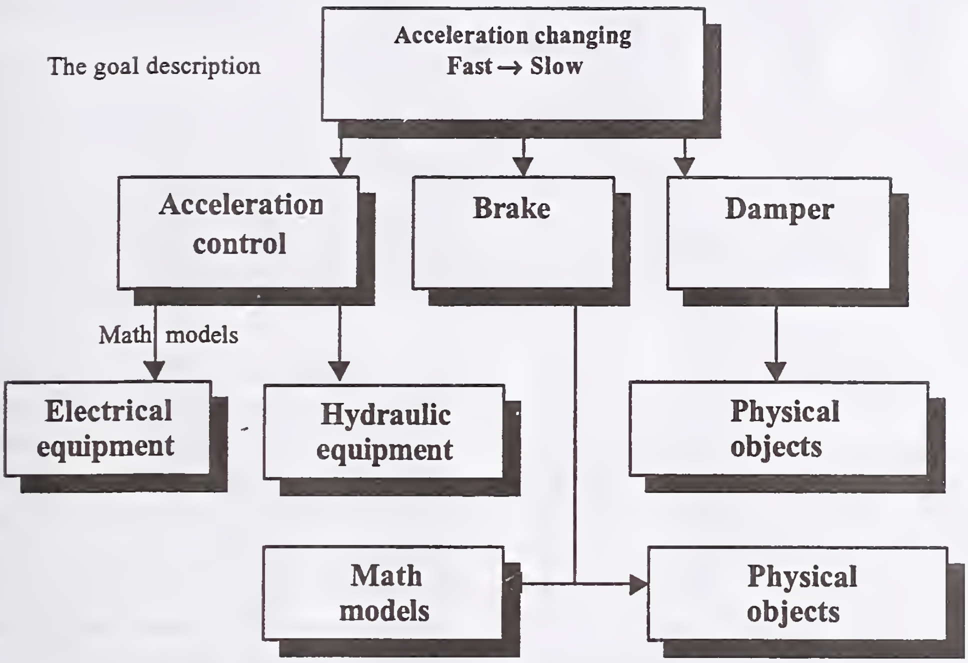


Fig. 4. Acceleration control system design (search through the goal hierarchy).

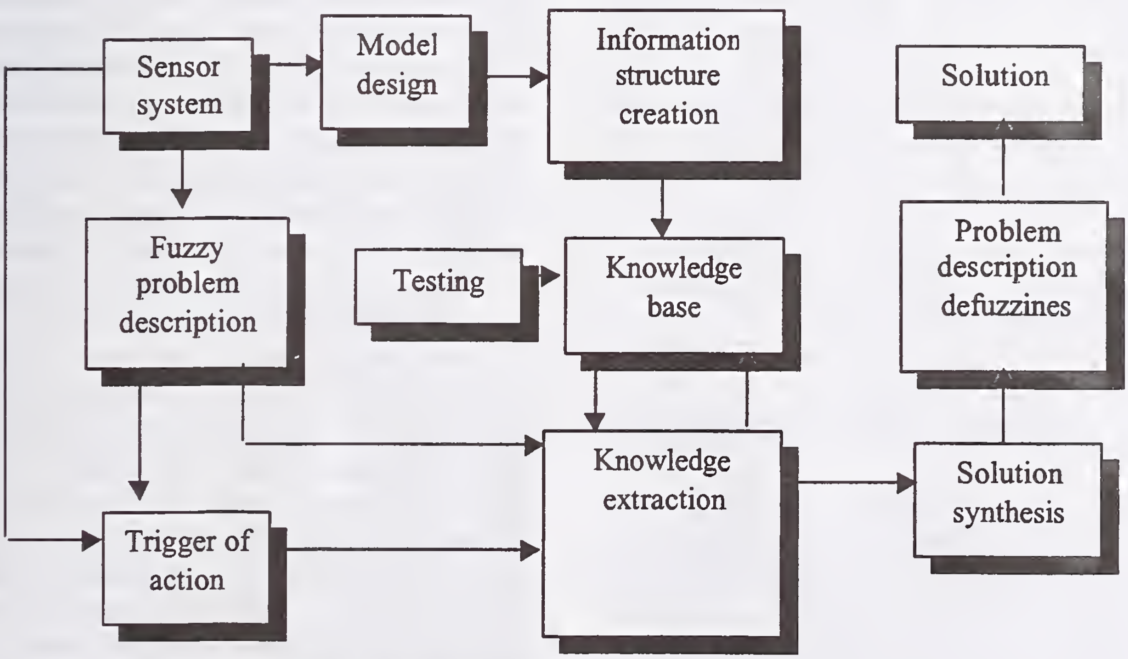


Fig. 5. The Artificial Intuition System Structure.

SEMIOTIC FUNDAMENTALS OF INFORMATION PROCESSING IN HUMAN BRAIN

Leonid I. Perlovsky

Air Force Research Lab., 80 Scott Rd., Hanscom AFB, MA 01731
Tel. 781-377-1728; e-mail: Leonid.Perlovsky@hanscom.af.mil

ABSTRACT

The paper discusses a mathematical nature of signs and symbols, and relates it to information processing and understanding, structure of the mind and brain, learning, and pattern recognition. I discuss past limitations of algorithms and neural networks, combina-torial complexity, the roles of concepts and emotions in mind's mechanisms, and various types of logic underlying mathematical techniques. A mathematical theory of semiosis, adaptive processes of sign interpretation, is described; it includes a similarity measure between signals and internal representations and fuzzy dynamic logic, a mechanism of the similarity maximization. Mathematical mechanisms of sign and symbol processing are presented and related to the functioning of mind.

KEYWORDS: *semiotics, symbols, fuzzy dynamic logic, neural networks, emotions, concepts, intelligent systems.*

1. SEMIOTICS, MIND, AND BRAIN

Semiotics studies signs and symbols, which are generally understood as entities designating some other entities in the world or in the mind. Using words like *mind, thought, imagination, emotion, concept* represents a specific challenge: people use these words in many ways colloquially, but their use in science and especially in mathematics of intelligence has not been uniquely defined and is a subject of active research and ongoing debates [1]. Whereas standardized definitions come at the end of the development of a theory (like "force" was defined by the 2nd Newton's law, following centuries of less precise usage) this paper adheres to a following guidance: we need to make sure that our definitions: (1) are mathematically exact, (2) correspond to the usage in scientific and mathematical community, (3) correspond to the general usage. According to a dictionary [2], mind includes conscious and unconscious processes, especially thought, perception, emotion, will, memory, and imagination, and it originates in brain. These constituent notions will be discussed throughout the paper. Specific neural mechanisms in brain "implementing" various mind functions constitute the relationship between the mind and brain; we will discuss how the mathematical descriptions of mind are implemented in brain.

In mathematics and in "Symbolic AI" there is no difference between signs and symbols. Both are considered as notations, arbitrary non-adaptive entities with axiomatically fixed meaning. But in general culture, symbols are understood also as psychological processes of sign interpretation. Jung emphasized that symbol-processes connect conscious and unconscious [3], Pribram wrote of symbols as adaptive, context-sensitive signals in the brain, whereas signs he identified with less adaptive and relatively context-insensitive neural signals [4].

In this paper I use "symbol" as a symbol-process, corresponding to general notions of symbol in culture and psychology. The symbol-processes are closely related to the processes of thinking, and a mathematical theory suitable for the description of sym-bols is closely related to the mathematical description of the working of the mind.

A broad range of opinions exists on the mathematical methods suitable for the description of the mind. Founders of artificial intelligence thought that formal logic was sufficient [5] and no specific mathematical techniques would be needed to describe the mind [6]. An opposite point of view is that there are few specific mathematical constructs, "the first principles" of mind. Among researchers taking this view is Grossberg, who suggests that the first principles include a resonant matching between lower-level signals [7] and higher-level representations and emotional evaluation of conceptual contents [8]; Zadeh develops theory of granularity [9], Meystel develops hierarchical multiscale organization with specific intra-level closed-loop structures [10]; and the author, suggests similarity measures between lower-level signals and higher-level representations [11] and the fuzzy dynamic logic [12] among first principles of mind.

2. MIND, LOGIC, AND COMPLEXITY

Understanding the meaning of signals coming from sensory organs involves associating the subsets of signals corresponding to an object with internal representations. This recognition activates internal brain signals leading to mental and behavioral responses involved in understanding.

Developing mathematical descriptions of the very first *recognition* step of this seemingly simple association-recognition-understanding process has not been easy, a number of difficulties have been encoun-tered during the past fifty years. These difficulties have been summarized under the term combinatorial complexity (CC) [11]. The problem was first identified in pattern recognition and classification problems in the 1960s and was named "the curse of dimensionality" [13]. The following thirty years of developing adaptive statistical pattern recognition and neural network algorithms led to a conclusion that these approaches often encountered *CC of learning requirements*. Rule-based systems were proposed to solve the problem of learning complexity. An initial idea was that rules would capture the required knowledge and eliminate a need for learning. However, rule systems and expert systems in the presence of variability, encountered *CC of rules*. Model-based systems were proposed to combine advantages of adaptivity and rules by utilizing adaptive models, but they encountered *computational CC* (N and NP complete algorithms).

Combinatorial complexity has been related to the type of logic, underlying various algorithms and neural networks [14]. Formal logic is based on the "law of excluded third", according to which every statement is either true or false and nothing in between. Therefore, algorithms based on formal logic have to evaluate every little variation in data or internal representations as a separate logical statement (hypothesis); a large number of combinations of these variations causes combinatorial complexity. In fact, combinatorial complexity of algorithms based on logic has been related to the Gödel theory: it is a manifestation of the incompleteness of logic in finite systems [15]. Multivalued logic and fuzzy logic were proposed to overcome limitations related to the law of excluded third [16]. Yet the mathematics of multivalued logic is no different in principle from formal logic. Fuzzy logic encountered a difficulty related to the degree of fuzziness, if too much fuzziness is specified, the solution does not achieve a needed accuracy, if too little, it might become similar to formal logic.

Another view on these difficulties can be obtained by comparing mathematical techniques to human mind. An essential role of emotions in the working of the mind was analyzed from the psychological and neural perspective by Grossberg [17], from the neurophysiological perspective by Damazio [18], and from the learning and control perspective by the author [19]. One reason for engineering community being slow in adopting these results is the cultural bias against emotions as a part of thinking processes. Plato and Aristotle thought that emotions are "bad" for intelligence, this is a part of our cultural heritage, and the founders of Artificial Intelligence repeated it. Yet, as discussed in the next section, combining conceptual understanding with emotional evaluations might be crucial for overcoming the combinatorial complexity as well as the related difficulties of logic.

3. MODELING FIELD THEORY (MFT)

Modeling field theory [11], summarized below, associates lower-level signals with higher-level representations, resulting in understanding of signals, while overcoming the difficulties described in the previous section. It is achieved by using flexible measures of similarity between the representations and the input signals combined with the fuzzy dynamic logic. Modeling field theory is a multi-level, hetero-hierarchical system. This section describes a basic mechanism of interaction between two adjacent hierarchical levels of signals (fields of neural activation); sometimes, it will be more convenient to talk about these two signal-levels as an input to and output from a (single) processing-level.

At each level, the output are concepts recognized (or formed) in input signals. Input signals **X** are associated with (or recognized, or grouped into) concepts according to the representations-models and similarity measures at this level. In the process of association-recognition, models are adapted for better representation of the input signals; and similarity measures are adapted so that their fuzziness is matched to the model uncertainty. The initial uncertainty of models is high and so .is the fuzziness of the similarity measure; in the process of learning models become more accurate and the similarity measure more crisp, the value of the similarity increases. We call this mechanism fuzzy dynamic logic.

3.1 Internal Models, Learning, and Similarity

During the learning process, new associations of input signals are formed resulting in evolution of new concepts. Input signals $\{\mathbf{X}(n),\ n \in N\}$, is a field of input neuronal synapse activation levels, $\mathbf{X} = \{X_d,\ d = 1,\ldots D\}$; a set of concepts $\{h \in H\}$ is characterized by internal parameters $\{\mathbf{S}_h\}$ and by models (representations) of the signals $\{\mathbf{M}_h(\mathbf{S}_h,n)\}$ corresponding to concepts $\{h\}$. For each model h, the set of parameters is denoted as $\mathbf{S}_h = \{S^a_h,\ a = 1,\ldots A\}$. Learning process increases a similarity measure between the sets of models and signals, $L(\{\mathbf{X}\},\{\mathbf{M}\})$. The similarity measure is a function of model parameters and associations between the input synapses and concepts-models. A similarity measure is designed so that it treats each model as an alternative for each subset of signals

$$L(\{\mathbf{X}\},\{\mathbf{M}\}) = \prod_{n \in N} \sum_{h \in H} r(h)\, l(\mathbf{X}(n) \mid h), \quad (1)$$

here $l(\mathbf{X}(n)|h)$ (or simply $l(n|h)$) is a conditional partial similarity between signal vector $\mathbf{X}(n)$ and model $\mathbf{M}_h$ (when mapping this terminology onto its implemetation in the brain, n and h are neural indexes numbering individual neurons or small groups of neurons). For example, $l(n|h)$ can be selected as a probability density function. Then L is a total likelihood (this interpretation does not require statistical independence among signal vectors n and n': dependencies are accounted for by model dependencies on {n}).

In the process of learning, concept-models are constantly modified. From time to time a system forms a new concept, while retaining an old one as well; alternatively, old concepts are sometimes merged. Formation of new concepts and merging of old ones require a modification of the similarity measure (1); the reason is that more models always result in a better fit between the models and data. This is a well known problem, it can be addressed by reducing (1) using a "penalty function", p(N,M) that grows with the number of models M, and this growth is steeper for a smaller amount of data N. For example, an asymptotically unbiased maximum likelihood estimation leads to multiplicative $p(N,M) = \exp(-N_{par}/2)$, where N_{par} is a total number of adaptive parameters in all models (this penalty function is known as Akaike Information Criterion, see [11] for further discussion and references).

In case, when a set of observations, N, corresponds to a continuous flow of signals, for example, a flow of visual stimuli in time and space, it is convenient instead of eq.(1) to consider its continuous version,

$$L = \exp ?_{N} \ln \left(\sum_{h \in H} r(h)\, l(\mathbf{X}(n) \mid h)\right), \qquad (2)$$

where N is a continuum, such as time-space. In this case, models describe continuous modeling fields and maximization of similarity L can be compared to minimization of action in a physical field theory.

3.2 Fuzzy dynamic logic and MFT

The learning process consists in estimating internal parameters S and associating subsets of signals with concepts by maximizing the similarity (1). When likelihood is used as a similarity measure, this is a problem of the maximum likelihood estimation. Note, that (1) contains a total of H^N items; this is a source of the combinatorial complexity in many algorithms (called maximum hypothesis testing) which attempt to maximize similar expressions by first maximizing each item over the parameters and then finding the maximal item.

Modeling field theory solves this problem by utilizing fuzzy dynamic logic [11,20]. Let us introduce association variables f(h|n)

$$f(h|n) = r(h)\, l(\mathbf{X}(n)|h) \,/ \sum_{h' \in H} r(h')\, l(\mathbf{X}(n)|h'). \qquad (3)$$

Eq.(3) looks like the Bayes formula for a posteriori probabilities, if l(n|h) are conditional likelihoods. An internal dynamics of the Modeling Fields (MF) is defined as follows,

$$df(h \mid n)/dt = f(h \mid n) \sum_{h' \in H} \{[\delta_{hh'} - f(h'|n)] \cdot$$

$$[?\ln l\,(n|h')/?\mathbf{M}_{h'}]\; \partial \mathbf{M}'_{h'}/\partial S_{h'} \cdot dS_{h'}/dt, \qquad (4)$$

$$dS_h/dt = \{?_{N} f(h|n)[?\ln l(n|h)/?\mathbf{M}_h] \partial \mathbf{M}'_h/\partial S_h, \qquad (5)$$

here

$\delta_{hh'}$ is 1 if h=h', 0 otherwise. (6)

Parameter t is the time of the internal dynamics of the MF system (like a number of internal iterations). A more specific form of (5) can be written when Gaussian-shape functions are used for conditional partial similarities,

$$l(n|h) = G(\mathbf{X}(n) \mid \mathbf{M}_h(S_h, n), \mathbf{C}_h). \qquad (7)$$

where G is a Gaussian function with mean $\mathbf{M}_h$ and covariance matrix $\mathbf{C}_h$ (this is not a necessary assumption, but it will simplify some discussions later, also, it is not same as usual Gaussian limitation, in fact, it is not much of a limitation at all, because a weighted sum of Gaussians in (1) can approximate any positive function). And let us specify the dynamics of the MFT as follows,

$$dS^{a}_{h}/dt = [Y_{h}^{-1}]^{ab} Z_{h}^{b}, \qquad (8)$$

$$dC_{h}/dt = -0.5\mathbf{C}_{h}^{-2} ?_{N} f(h|n)[\mathbf{C}_{h}-\mathbf{D}_{nh}\,\mathbf{D}_{nh}^{T}]; \qquad (9)$$

$$\mathbf{D}_{nh} = (\mathbf{X}(n) - \mathbf{M}_{h'}), \qquad (10)$$

$$Y_{h}^{ab} = ?_{N} f(h|n)[\mathbf{M}^{;a}_{h}\mathbf{C}_{h}^{-1}\mathbf{M}^{;b}_{h}], \qquad (11)$$

$$Z_{h}^{b} = ?_{N} f(h|n)[\mathbf{M}^{;b}_{h}\mathbf{C}_{h}^{-1}\mathbf{D}_{nh}], \qquad (12)$$

here superscript T denotes a transposed row-vector; summation is assumed over repeated indexes a, b; and (;) denotes partial derivatives with respect to parameters S with corresponding indexes:

$$\mathbf{M}^{;b}_{h} = \partial \mathbf{M}_{h} / \partial S^{b}_{h}. \qquad (13)$$

The following theorem was proven.

Theorem. Equations (3) through (6) (or (3) and (8 through 12)) define a convergent dynamic system MF with stationary states defined by $\max_{\{S_h\}} L$.

It follows that the stationary states of an MF system give the maximum similarity solution of the model-based pattern recognition problem. When likelihood is used as similarity, the stationary values of parameters $\{S_h\}$ are asymptotically unbiased and efficient estimates of these parameters [21]. A computational complexity of the MF method is linear in N.

3.3 MFT hierarchical organization

The previous sub-section described a single processing layer in a hierarchical MFT system. An input to each layer is a set of signals **X**(n), or in neural terminology, an input field of neuronal activations. An output are the activated models $\mathbf{M}_h(\mathbf{S}_h, n)$; it is a set of models or concepts recognized in the input signals. Equations (3-6) or (3) and (7-12) describe a loop-process: at each iteration (or internal-time t) the l.h.s. of the equations contain association variables f(h|n) and other model parameters computed at the previous iteration. In other words, the output models "act" upon the input to produce a "refined" output models (at the next iteration). This process is directed at increasing the similarity between the models and signals. It can be described as an internal behavior generated by the models.

The output models initiate other actions as well. First, activated models (neuronal axons) serve as input signals to the next processing layer, where more general concept-models are recognized or created (internal behavior within the MFT system). Second, concept-models along with the corresponding instinctual signals and emotions may activate behavioral models and generate behavior directed into the outside world (a process not contained within the above equations).

MFT describes an intelligent system composed of multiple adaptive intelligent agents: each concept-model is an agent, which is "dormant" until activated by a high similarity value. When activated, it is adapted to the signals, so that the similarity increases. Every piece of signal may activate several concepts, which "compete" with each other, while adapting to the new signals.

3.4 MFT theory of mind

MFT dynamics, (3) and (4-6) or (7-12), describes an elementary process of perception or cognition, in which a large number of model-concepts compete for incoming signals, model-concepts are modified and new ones are formed, and eventually, connections are established among signal subsets on the one hand, and model-concepts on the other. Perception refers to processes in which the input signals come from sensory organs and model-concepts correspond to objects in the surrounding world. Cognition refers to higher levels in the hierarchy where the input signals are concepts activated at lower levels and model-concepts are more complex and correspond to situations and relationships among lower-level concepts.

A salient mathematical property of this processes ensuring a smooth convergence is a correspondence between uncertainty in models (that is, in the knowledge of model parameters) and uncertainty in associations f(h|n). In perception, as long as model parameters do not correspond to actual objects, there is no match between models and signals; many models poorly match many objects, and associations remain fuzzy; this can be described more specifically, if Gaussian functions are used for l(**X**|h): for poorly matched models, the covariances, $\mathbf{C}_h$, are large (that is, model uncertainties are large), which in turn prevents f(h|n) from attaining definite (0,1) values. Eventually, one model (h') wins a competition for a subset {n'} of input signals **X**(n), when parameter values match object properties, $\mathbf{C}_{h'}$ becomes smaller than other $\mathbf{C}_h$, and f(h'|n) values become close to 1 for $n \in \{n'\}$ and 0 for $n \notin \{n'\}$. Upon the convergence, the entire set of input signals {n} is divided into subsets, each associated with one model-object, $\mathbf{C}_h$ become small, and fuzzy a priori concepts become crisp concepts. Cognition is different from perception in that models are more general, more abstracts, and input signals are the activation signals from concepts identified (cognized) at a lower hierarchical level; the general mathematical laws of cognition and perception are similar in MFT. Let us discuss relationships between the MFT theory, theory of solitons in non-linear systems and concepts of mind originated in psychology, philosophy, linguistics, aesthetics, neuro-physiology, neural networks, artificial intelligence, pattern recognition, and intelligent systems.

Solitons and MFT resonances. The physical nature of concepts of mind in MFT is similar to that of solitons. If the data **X**(n) are all given from the very beginning, equations (3-6) or (7-12) converge to a fixed point of MFT system. This fixed point is comprised of a number of resonances [22] between the field of models and field of data, in other words, the models come into a resonance with the data, and the system stays in this resonant state. Formation of a resonance takes different time (number of iterations) for various models, and it is more proper to talk about each model coming into a resonance with a corresponding data subset. If there is a continuous flow of data, **X**(n,t), a resonance is a long-living state (long comparative to a single iteration cycle). The nature of this resonance between the modeling fields and the data field is such that a particular subset of data (corresponding to an object h) "drives" the modeling-field to a specific value (or pattern) $\mathbf{M}_h(\mathbf{S}_h$,n,t), and these modeling-field values "drive" the association-fields, f(h|n), to {0,1} values. It follows that concepts of mind in MFT theory are resonant states, or solitons of a highly nonlinear MFT system. It is interesting to note recent results [23] establishing relationships between solitons in certain nonlinear systems and theorems of inversive geometry. More research is needed to establish general relationships between concepts of mind as long-living resonant states in a nonlinear system and a body of results obtained in the theory of integrable systems and solitons [24].

Elementary thought-process, conscious-ness, and unconscious. A thought-process or thinking involves a number of sub-processes and attributes, including internal representations and their manipulation, attention, memory, concept formation, knowledge, generalization, recognition, understanding, meaning, prediction, imagination, intuition, emotion, decisions, reasonining, goals, behavior, conscious and unconscious [7,10,11].

A "minimal" subset of these processes has to involve mechanisms for afferent and efferent signals [22], in other words, bottom-up and top-down signals coming from outside (external sensor signals) and from inside (internal representation signals). According to Carpenter and Grossberg [22] every recognition and concept formation process involves a "resonance" between these two types of signals. In MFT, at every level in a hierarchy the afferent signals are represented by the input signal field **X**, and the efferent signals are represented by the modeling field signals $\mathbf{M}_h$; resonances correspond to high similarity measures l(n|h) for some subsets of {n} that are "recognized" as concepts (or objects) h. The mechanism leading to the resonances is given by (3-6) or (7-12), and we call it an elementary thought-process. The elementary thought-process involves elements of conscious and unconscious processes, imagination, memory, internal representations, concepts, instincts, emotions, understanding and behavior as further described later.

A description of working of the mind as given by the MFT dynamics was first provided by Aristotle [25], describing thinking as a learning process in which an a priori form-as-potentiality (fuzzy model) meets matter (sensory signals) and becomes a form-as-actuality (a concept). Jung suggested that conscious concepts are developed by mind based on genetically inherited structures of mind, archetypes, which are inaccessible to consciousness [3]; and Grossberg [7] suggested that only signals and models attaining a resonant state (that is signals matching models) reach consciousness.

Understanding. In the elementary thought process, subsets in the incoming signals are associated with recognized model-objects, creating *phenomena* (of the MFT-mind) which are *understood* as objects, in other words *signal subsets* acquire *meaning* (e.g., a subset of retinal signals acquires a meaning of a chair). There are several aspects to understanding and meaning. First, object-models are connected (by emotional signals [8,11,19]) to instincts that they might satisfy, and also to behavioral models that can make use of them for instinct satisfaction. Second, an object is understood in the context of a more general situation in the next layer consisting of more general concept-models, which accepts as input-signals the results of object recognition. That is, each recognized object-model (phenomenon) sends (in neural terminology, activates) an output signal; and a set of these signals comprises input signals for the next layer models, which 'cognize' more general concept-models. And this process continues up and up the hierarchy of models and mind toward the most general models a system could come up with, such as models of universe (scientific theories), models of self (psychological concepts), models of meaning of existence (philosophical concepts), models of a priori transcendent intelligent subject (theological concepts).

Imagination. Imagination involves excitation of a neural pattern in a visual cortex in absense of an actual sensory stimulation (say, with closed eyes) [7]. Imagination was often considered to be a part of thinking processes; Kant [26] emphasized the role of imagination in the thought process, he called thinking "a play of cognitive functions of imagination and understanding". Whereas pattern recognition and artificial intelligence algorithms of recent past would not know how to relate to this [5,6], Carpenter and Grossberg resonance model [22] and the MFT dynamics both describe imagination as an inseparable part of thinking: imagined patterns are top-down signals that *prime* the percepting cortex areas (*priming* is a neural terminology for making neural cells to be more readily excited). In MFT, the imagined neural patterns are given by models $\mathbf{M}_h$. MFT (in agreement with neural data) just adds details to Kantian description: thinking is a play of *higher-hierarchical-level* imagination and *lower-level* understanding. Kant identified this "play" [described by (3-6) or (7-12)] as a source of aesthetic emotion; modeling aesthetic emotion in MFT is described later.

Mind vs. Brain. Historically, the mind is described in psychological and philosophical terms, whereas the brain is described in terms of neurobiology and medicine. Withing scientific exploration the mind and brain are different description levels of the same system. Establishing relationships between these description is of great scientific interest. Today we approach solutions to this challenge [[27]], which eluded Newton in his attempt to establish physics of "spiritual substance" [[28]]. General neural mechanisms of the elementary thought process (which are similar in MFT and ART [[22]]) have been confirmed by neural and psychological experiments, this includes neural mechanisms for bottom-up (sensory) signals, top-down "imagination" model-signals, and the resonant matching between the two [[29]]. Adaptive modeling abilities are well studied with adaptive parameters identified with synaptic connections [[30]]; instinctual learning mechanisms have been studied in psychology and linguistics [[31]].

Instincts and emotions. Functioning of the mind and brain cannot be understood in isolation from the system's "bodily needs". For example, a biological system (and any autonomous system) needs to replenish its energy resources (eat); this and other fundamental unconditional needs are indicated to the system by instincts, which could be described as internal sensors. Emotional signals, generated by this instinct are perceived by consciousness as "hunger", and they activate behavioral models related to food searching and eating. In this paper we are concerned primarily with the behavior of recognition: instinctual influence on recognition modify the object-perception process (3) - (6) in such a way, that desired objects "get" enhanced recognition; it can be accomplished by modifying priors, r(h).

Aesthetic emotions and instinct for knowledge. Recognizing objects in the environment and understanding their meaning is so important for human evolutionary success that there has evolved an instinct for learning and improving concept-models. This instinct is described in MFT by maximization of similarity between the models and the world, (1). Emotions related to satisfaction-dissatisfaction of this instinct are perceived by us as harmony-disharmony (between our understanding of how things ought to be and how they actually are in the surrounding world). According to Kant [[32]] these are aesthetic emotions.

Intuition includes an intuitive perception (imagination) of object-models and their relationships with objects in the world, as well as higher-level models of relationships among simpler models. Intuition involves fuzzy unconscious concept-models, which are in a state of being learned and being adapted toward crisp and conscious models (a theory); such models may satisfy or dissatisfy the knowledge instinct in varying degrees before they are accessible to consciousness, hence the complex emotional feel of an intuition. The beauty of a physical theory discussed often by physicists is related to satisfying our feeling of purpose in the world, that is, satisfying our need to improve the models of the meaning in our understanding of the universe.

Beauty. Harmony is an elementary aesthetic emotion related to improvement of object-models. Higher aesthetic emotions are related to the development of more complex "higher" models: we perceive an object or situation as aesthetically pleasing if it satisfies our learning instinct, that is the need for improving the models and increasing similarity (1). The highest forms of aesthetic emotion are related to the most general and most important models. According to Kantian analysis [[32]], among the highest models are models of the meaning of our existence, of our purposiveness or intentionality, and beauty is related to improving these models: we perceive an object or a situation as beautiful, when it stimulates improvement of these highest models of meaning. Beautiful is what “reminds” us of our purposiveness.

Theory testing and future directions. The general neural mechanisms of the elementary thought process, which includes neural mechanisms for bottom-up (sensory) signals, top-down “imagination” model-signals, and the resonant matching between the two [33], have been confirmed by neural and psychological experiments (these mechanisms are similar in MFT and ART [22]). Adaptive modeling abilities are well studied and adaptive parameters have been identified with synaptic connections [34]; instinctual learning mechanisms have been studied in psychology and linguistics [35]. Ongoing and future research will confirm, disprove, or suggest modifications to specific mechanisms of model parameterization and parameter adaptation (5) or (8), reduction of fuzziness during learning (9), similarity measure (1) as a foundation of aesthetic instinct for knowledge, relationships between psychological and neural mechanisms of learning on the one hand and, on the other, aesthetic feelings of harmony and emotion of beautiful. Differentiated forms of (1) need to be developed for various forms of the knowledge instinct (child development, language learning, etc.) Future experimental research needs to study in details the nature of hierarchical interactions: to what extent the hierarchy is “hardwired” vs. adaptively emerging; what is a hierarchy of learning instinct? theory of emerging hierarchical models will have to be developed (that is, adaptive, dynamic, fuzzy hierarchy- heterarchy).

4. THINKING PROCESS AND SEMIOTICS

Semiotics studies symbol-content of culture [36]. For example, consider a written word "chair". It can be interpreted by a mind to refer to something else: an entity in the world, a specific chair, or the concept "chair" in the mind. In this process, the mind, or an intelligent system is called *an interpreter*, the written word is called *a sign*, the real-world chair is called *a designatum,* and the concept in the interpreter's mind, the internal representation of the results of interpretation is called *an interpretant* of the sign. The essence of a sign is that it can be interpreted by an interpreter to refer to something else, a designatum. This process of sign interpretation is an element of a more general process called semiosis which consists of multiple processes of sign interpretation at multiple levels of the mind hierarchy.

In classical semiotics [37] words *sign* and *symbol* were not used consistently; in this paper, a sign means something that can be interpreted to mean something else (like a mathematical notation, or a word), and the process of interpretation is called a symbol-process, or symbol. Intepretation, or understanding of a sign by the mind according to MFT is due to the fact that a sign (e.g., a word) is a part of an object-model (or a situation-model at higher levels of the mind hierarchy). The mechanism of a sign interpretation therefore involves first an activation of an object-model, which is connected to instincts that the object might satisfy, and also to behavioral models that can make use of this object for instinct satisfaction. Second, a sign is understood in the context of a more general situation in the next layer consisting of more general concept-models, which accepts as input-signals the results of lower-level sign recognition. That is, recognized signs comprise input signals for the next layer models, which ‘cognize’ more general concept-models.

A symbol-process of a sign interpretation coincides with an elementary thought-process. Each sign-interpretation or elementary thought process, a symbol, involves conscious and unconscious, emotions and concepts; this definition connecting symbols to archetypes (fuzzy unconscious model-concepts) corresponds to a usage in general culture and psychology [3,4]. As described previously, this process continues up and up the hierarchy of models and mind toward the most general models. In semiotics this process is called *semiosis*, a continuous process of creating and interpreting the world outside (and inside our mind) as an infinite hierarchical stream of signs and symbol-processes.

REFERENCES

[1] For the discussions and further references see: Grossberg, S. (1988). *Neural Networks and Natural Intelligence.* MIT Press, Cambridge, MA. Albus, J.S., Meystel, A.M. (2001). *Engineering of Mind: An Introduction to the Science of Intelligent Systems.* Wiley, New York, NY. Meystel, A.M., Albus, J.S. (2001). *Intelligent Systems: Architecture, Design, and Control.* Wiley, New York, NY. Perlovsky, L.I. (2001). *Neural networks and intellect: using model based concepts.* Oxford University Press, NY.

[2] The American Heritage College Dictionary, Houghton Mifflin, 3rd ed., 2000, Boston, MA.

[3] Jung, C.G. (1934). *Archetypes of the Collective Unconscious.* In the Collected Works, v.9,II, Bollingen Series XX, 1969, Princeton University Press, Princeton, NJ.

[4] Pribram,K. (1971). Languages of the Brain. Prentice Hall.

[5] Newell, A. (1983). Intellectual issues in the history of artificial intelligence. In The study of information, ed. F.Machlup & U.Mansfield, J.Wiley, New York, NY.

[6] Minsky, M. (1988). The Society of Mind. MIT Press, Cambridge, MA.

[7] Grossberg, S. (1988). *Neural Networks and Natural Intelligence.* MIT Press, Cambridge, MA.

[8] Grossberg, S. & Levine, D.S. (1987). Neural dynamics of attentionally modulated Pavlovian conditioning: blocking, inter-stimulus interval, and secondary reinforcement. *Psychobiology*, **15**(3), pp.195-240;

[9] Zadeh, L.A. (1997). *Information granulation and its centrality in human and machine intelligence.* Proceedings of the Conf. on Intelligent Systems and Semiotics '97. Gaithersburg, MD, pp. 26-30.

[10] Meystel, A. (1995). Semiotic Modeling and Situational Analysis. AdRem, Bala Cynwyd, PA.

[11] Perlovsky, L.I. (2001). *Neural networks and intellect: using model based concepts.* Oxford University Press, NY.

[12] Fuzzy dynamic logic is discussed mathematically below, it is an extension of fuzzy logic of Zadeh, and it is not directly related to other techniques of "dynamic logic".

[13] Bellman, R.E. (1961). *Adaptive Control Processes.* Princeton University Press, Princeton, NJ.

[14] Perlovsky, L.I. (1998). *Conundrum of Combinatorial Complexity.* IEEE Trans. PAMI, **20**(6) p.666-70.

[15] Perlovsky, L.I. (1996). *Gödel Theorem and Semiotics.* Proceedings of the Conference on Intelligent Systems and Semiotics '96. Gaithersburg, MD, v.2, pp. 14-18.

[16] Jang,J.-S.R., Sun,C.-T., and Mizutani, E. (1996). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall, Upper Saddle River, NJ.

[17] Grossberg, S.&Schmajuk, N.A. (1987). Neural dynamics of attentionally modulated Pavlovian conditioning: Conditioned reinforcement, inhibition, and opponent processing.*Psychobiology*,**15**(3),p.195-240.

[18] Damasio, A.R. (1995). Descartes' Error: Emotion, Reason, and the Human Brain. Avon, NY, NY.

[19] Perlovsky, L.I. (1998). *Cyberaesthetics: aesthetics, learning, and control.* STIS'98, Gaithersberg, MD; Perlovsky, L.I. (1999). *Emotions, Learning, and Control.* Proc. International Symp. Intelligent Control, Intelligent Systems & Semiotics, Cambridge MA, pp.131-137

[20] Perlovsky, L.I. (1996). *Mathematical Concepts of Intellect.* Proc. World Congress on Neural Networks, San Diego, CA; Lawrence Erlbaum Associates, NJ, pp.1013-16; Perlovsky, L.I.(1997).*Physical Concepts of Intellect.* Proc. Russian Academy of Sciences, **354**(3), pp. 320-323.

[21] Cramer, H. (1946). *Mathematical Methods of Statistics,* Princeton University Press, Princeton NJ.

[22] G.A.Carpenter & S.Grossberg, 1987, A massively parallel architecture for a self-organizing neural pattern recognition machine, *Computer Vision, Graphics and Image Processing*, **37**, 54-115.

[23] B.G. Konopelchenko & W.K. Schief, 2001, Reciprocal figures, graphical statics and inverse geometry of the Schwarzian BKP hierarchy, will be published in Studies in Applied Mathematics (2002); B.G. Konopelchenko & W.K. Schief, 2001, Menelau's theorem, Clifford configurations and inverse geometry of the Schwarzian KP hierarchy, to be published.

[24] V.E. Zakharov, 1991, *What Is Integrability?* Springer-Verlag Wien NewYork.

[25] Aristotle, IV BC, *Metaphysics,* tr. W.D.Ross, Complete Works of Aristotle, Ed.J.Barnes, Princeton, NJ, 1995.
[26] Kant, I. (1790). *Critique of Judgment,* tr. J.H.Bernard, Macmillan & Co., London, 1914.
[27] Grossberg, S. (2000). Linking mind to brain: the mathematics of biological intelligence. Notices of the American Mathematical Society, 471361-1372.
[28] Westfall, R.S. (1983). Never at Rest: A Biography of Isaac Newton. Cambridge Univ Pr., Cambridge.
[29] S.Grossberg, *Neural Networks and Natural Intelligence.* MIT Press, Cambridge, MA, 1988, S.Zeki, *A Vision of the Brain* Blackwell, Oxford, England, 1993; W.J. Freeman, *Mass action in the nervous system.* Academic Press, New York, NY, 1975.
[30] C.Koch and I.Segev, Edts., *Methods in Neuronal Modeling: From Ions to Networks* MIT Press, Cambridge, MA 1998; D.Hebb, *Organization of Behavior* J.Wiley & Sons, New York, NY, 1949.
[31] J.Piaget, *The Psychology of the Child* Tr. H.Weaver, Basic Books, 2000); N.Chomsky, in Explanation in Linguistics, ed. N.Hornstein and D.Lightfoot Longman, London, 1981. Jackendoff, R. (2002). Foundations of Language : Brain, Meaning, Grammar, Evolution, Oxford Univ Pr.; Deacon, T.W. (1998). The Symbolic Species : The Co-Evolution of Language and the Brain. W.W. Norton & Company.
[32] Kant, I. (1790). *Critique of Judgment,* tr. J.H.Bernard, Macmillan & Co., London, 1914.
[33] S.Grossberg, *Neural Networks and Natural Intelligence.* MIT Press, Cambridge, MA, 1988, S.Zeki, *A Vision of the Brain* Blackwell, Oxford, England, 1993; W.J. Freeman, *Mass action in the nervous system.* Academic Press, New York, NY, 1975.
[34] C.Koch and I.Segev, Edts., *Methods in Neuronal Modeling: From Ions to Networks* MIT Press, Cambridge, MA 1998; D.Hebb, *Organization of Behavior* J.Wiley & Sons, New York, NY, 1949.
[35] J.Piaget, *The Psychology of the Child* Tr. H.Weaver, Basic Books, 2000); N.Chomsky, in Explanation in Linguistics, ed. N.Hornstein and D.Lightfoot Longman, London, 1981.
[36] Sebeok, T.A. 1995, *Sign: An Introduction to Semiotics.* Univ. Toronto Press, Toronto, Canada.
[37] Peirce, C.S. (1935-66). *Collected Papers of Charles Sanders Peirce.* Harvard Univ. Press., Cambridge, MA; Morris C. (1971). Writings on the General Theory of Signs, ed. Th. A. Sebeok. The Hague: Mouton.

Intelligence and Behavioral Boundaries

Scott A. Wallace and John E. Laird
Artificial Intelligence Laboratory
University of Michigan
Ann Arbor, MI 48109, USA
{swallace,laird}@umich.edu

ABSTRACT

In this paper, we examine Newell's definition of an agent's intelligence. The definition implies a very strict correlation between the agent's knowledge, its mind, and its behavior. Based on this correlation we argue that a concise and well-formed representation for the agent's behavior is often essential for measuring its intelligence. To meet this need, we present Behavioral Bounding, a method that can be used to calculate a concise, high-level representation of behavior. We examine both the theoretical commitments of this approach, as well as practical limitations.

KEYWORDS: *version-spaces, behavior representations*

1 INTRODUCTION

In his 1990 book, *Unified Theories of Cognition* (UTC) [6], Newell observes that notions of intelligence are often linked to an underlying service to a particular cause. In this sense, there is no single universal concept of intelligence, but rather a multitude of intelligence measures each related to a particular field, task, or area of interaction. This idea gives rise to such common expressions as *academic intelligence* and *real-world intelligence*. As Newell indicates, the value of an intelligence metric is presumably to help identify which minds can perform which tasks, and what their relative abilities might be.

As a foundation for talking about intelligent agents, Newell introduces the notion of a *knowledge-level* system [5, 6]. As he describes it, such a system is located within an environment and performs a series of actions to obtain its goals; it selects between potential actions by using all of its relevant knowledge [6, pp. 50]. From this concept, Newell constructs the following definition of intelligence: "A system is *intelligent* to the extent that it approximates a knowledge-level system" [6, pp. 90]. That is, a system is perfectly intelligent if it uses all of its available knowledge to achieve its goals. Furthermore, lack of knowledge is distinct from a lack of intelligence. Thus, it often makes most sense to compare the intelligence of two systems with respect to a explicit body of knowledge and a specific problem domain as when one assess the impact of street-smarts on academic activities.

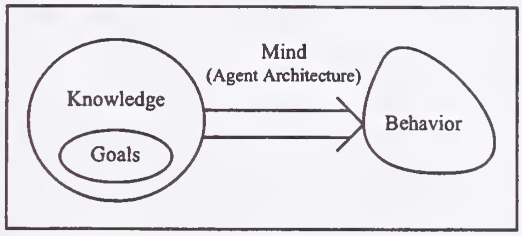


Figure 1: A Knowledge-Level System

Because a true knowledge-level system uses its knowledge to generate behavior that achieves its goals, the system's behavior should be predictable (up to aspects of behavior about which the agent is indifferent) so long as its knowledge is completely known (see Figure 1). A system that approximates the knowledge level, however, is one that cannot reliably make use of all its knowledge. Prediction of this system's behavior requires knowing what the system knows, as well as how its mind works (i.e. the implementation details governing how knowledge is brought to bear, in essence the agent architecture).

The relationship between knowledge, mind and behavior means that given information about two of these three components, it is possible to make a reasonable hypothesis about the third component. In this manner, one can determine how well an agent approximates the knowledge-level, how *intelligent* it is, so long as the agent's knowledge and its behavior are known.

2 BEHAVIOR & INTELLIGENCE

In some situations, it may be difficult or impossible to completely and correctly specify the agent's knowledge. This may be because the agent cannot be trusted to accurately communicate what it knows. Alternatively, it may be because it is exceedingly difficult to interpret the agent's native encoding of knowledge. In either of these two situations, it seems reasonable to rely on the knowledge engineer, who designed the agent, to provide a specification for what the agent knows. Unfortunately, the knowledge engineer's response may be misleading. This is because the agent's knowledge may not have been fully validated and thus may differ from the developers specification (i.e. it may contain errors).

If it is impossible to completely and correctly specify the agent's knowledge, it is also impossible to apply Newell's definition of intelligence in its strict sense. However, we may still profit from an approximate measure of intelligence based on beliefs about what the agent knows. Since these beliefs need not be completely accurate, they should be relatively easy to obtain. Beliefs about the agent's knowledge may come from information offered by the agent itself or from the designer of that agent. Given this approximate information about the agent's knowledge, we can obtain an approximate measure of its intelligence by observing its behavior.

Regardless of whether we have perfect information about the agent's knowledge, making an inference about the agent's intelligence requires the ability to classify and represent behavior in a concise, and well-formed structure. In this paper, we will outline a technique for this purpose. This technique, which we call Behavioral Bounding (B-Bounding), allows us to construct a concise representation of how an agent may perform in various situations. Once this representation has been computed, it can be used to: obtain a true measure of the agent's intelligence (given complete information about the agent's knowledge); to compare relative approximate intelligence (between two agents that are believed to have the same knowledge); or validate an agent's knowledge base (given complete information about the capabilities of the agent's mind). In this paper, we will concern ourselves with the first two applications.

3 MEASURING INTELLIGENCE

As we outlined in Section 2, we can generate an approximate measure of a system's intelligence given a set of beliefs about that system's knowledge and a set of observations about its behavior. The basic process for determining the system's intelligence is given by Algorithm 1.

Algorithm 1 Measure Intelligence

```
B ← Beliefs About Agent's Knowledge
O ← Sequences of Observed Agent Behavior
I ← Max Intelligence
for all observed (state,action) sequence o in O do
  if o can be explained using B then
    do nothing
  else
    decrease I in a meaningful way[1]
  end if
end for
return I
```

This algorithm will return a measure of the agent's intelligence given any observed behavior and any set of beliefs about the agent's knowledge. The exact nature of this algorithm's result, however, is closely tied to the quality of input.

The first parameter of the algorithm is B, a set of beliefs about the agent's knowledge. The accuracy of the measurement will depend on how closely these beliefs reflect the agent's actual knowledge. If, for example, the beliefs turn out to have nothing in common with the agent's actual knowledge, none of the agent's actions will be explainable, and so the agent will be presumed to have a very low (or zero) intelligence, regardless of whether it is a true knowledge-level system. On the other hand, if the beliefs correspond exactly to the agent's true knowledge, then any incorrect behavior will be appropriately ascribed to a lack of intelligence.

The second parameter of the algorithm, O, is a set of $(state_0, action_0) \ldots (state_n, action_n)$ sequences describing behavior the agent was observed performing. This parameter affects the generality of the measurement for the agent's intelligence. If the scope of action sequences spans a large number of tasks and the agent's behavior is diverse, the measurement of intelligence is likely to be more general than if the agent was observed only within a very narrow context. When determining what agent to select for a particular task, one would like the most intelligent agent with respect to the knowledge needed to perform that task. Furthermore, if the task is well specified, the agent's intelligence in areas known to be outside of the

[1]There are many possible methods of decreasing the agent's intelligence score. Naively, one could set $I_{max} = 100$, and for each unexplained action decrease I by $1/N$ where $N =$ *total number of explanations attempted* thus producing a normalized measure. A discussion of the consequences of different approaches, however, is outside the scope of this paper.

task's score can be ignored. On the other hand, if the exact nature of the task is ill–defined, it may be important that the agent have a high general intelligence, even if it is less intelligent on some specific subset of tasks.

The simple algorithm we have presented allows the flexibility to compute measurements of intelligence that span different degrees of accuracy and generality. However, it does suffer from two significant drawbacks if it is used to compute a relatively general measure of intelligence.

Most importantly, Algorithm 1 requires explaining all the actions in each sequence of observed agent behavior. As the generality of the intelligence measure increases, the number of agent traces used by the algorithm will also increase. Since the cost of explaining each observation of behavior is likely to be very high, this could make calculating certain measurements of intelligence infeasible.

In addition, Algorithm 1 does not have any built in method to determine the breadth of behavior that was examined to compute the agent's intelligence. As a result, it may be unclear what types of tasks the measurement pertains to. Furthermore, in cases where the observations of agent behavior are similar, we would like to be able to exploit information acquired during previous explanations.

Improvements to the simple algorithm would reduce the cost of general measures of intelligence, and provide an indication of the space of tasks or situations to which the measurement can be applied. In the remaining sections of the paper, we describe the Behavioral Bounding method, and how it can be used to overcome these shortcomings.

4 BEHAVIORAL BOUNDING

The Behavioral Bounding method can be used to overcome the two faults with the basic *Measure Intelligence* algorithm described in Section 2. The main idea behind the methodology is to construct a concise high–level representation of an agent's behavior from a set of observations. This new representation allows two new methods of measuring general intelligence without explaining all of the behavior in each observation. In addition, the representation itself can be used to obtain a measure of the diversity of behavior encapsulated in the observed agent actions.

Conceptually, Behavioral Bounding is very similar to Mitchell's Version Space framework [4]. Given a language to describe an agent's behavior, and a general to specific ordering over representations in this language, we can construct a maximally specific representation of the observed agent behavior. Figure 2 illustrates this concept. Each

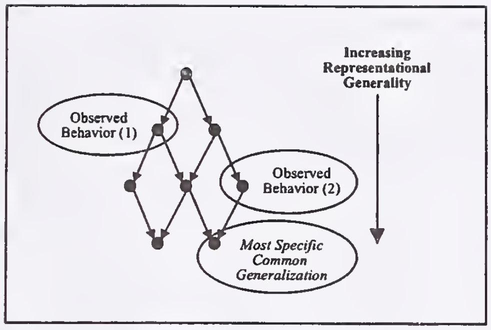


Figure 2: Ordered Behavior Representations

node corresponds to an abstract representation of behavior. Nodes toward the top of the lattice are more specific representations than nodes toward the bottom. Each of the observed agent behaviors is mapped onto this lattice, and the most specific common generalization can then be used to represent all of the behaviors that have been observed. Figure 2 illustrates a specific case with two observations of the agent's behavior. The abstract representation covering both of these observations is their common descendant.

Behavioral Bounding allows two new methods of measuring intelligence, both of which have an improvement in performance over our original simple algorithm. In the first method, the abstract representation of the agent's behavior is computed using all the available observations. Next, we substitute this abstract representation of behavior for the set of observations, O, in the original algorithm and proceed with the calculation in an otherwise normal manner. The benefit of this approach is that the number of explanations is no longer strictly a function of the number of observations. However, in practice this method may be difficult to apply. This is because the induced behavioral representation may be overly general, and as a result may encapsulate behaviors that the agent cannot explain. In such situations, the measurement of the agent's intelligence may be incorrect.

The second variation on our original algorithm overcomes this flaw. This approach uses the generalized behavior representation to tune the simple algorithm without actually substituting it for the observed agent behavior O. As in our first modification to the simple algorithm, we begin by constructing a general representation of the agent's behavior. Just as before, we initialize this representation, R_s, with the most specific representation of all possible behavior (this corresponds to initializing the S–SET in standard version space). Then, each observed behavior trace $o \in O$ is used to iteratively generate a new general representation, R'_s, of o and R_s. In our first variation, this was done by simply setting $R_s \leftarrow R'_s$ and repeating for all $o \in O$, thereby obtaining a generalization of all the traces in O. In this second approach, however, we examine the differences between R'_s and R_s before proceeding with the generalization. The aspects of R'_s that are more general than R_s indicate aspects of o that must be explained, whereas aspects of behavior that remain unchanged between R_s and R'_s have already been explained in a previous iteration. In this way the modified algorithm performs the minimal amount of explication for any set of observations.

The second variation of our original algorithm hints as to how the generalized behavior representation can be used to measure the diversity of observed agent behavior. After viewing a single instance of agent behavior, R_s will be generalized to the most specific representation that covers that behavior. Given progressively more instances of behavior, R_s will be generalized the minimal amount necessary to cover these observations. A long series of observations in which the agent's behavior is relatively similar will result in very few generalizations. Thus, differences (in terms of generality) between the most specific behavior representation and the behavior representation that covers all of the observations provides an automatic way of determining the generality of the intelligence measurement.

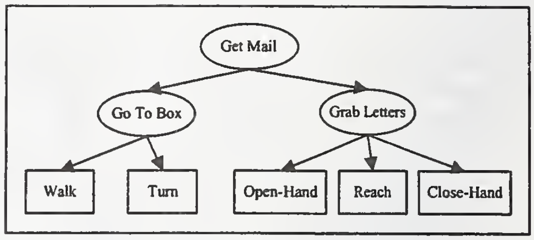


Figure 3: Hierarchical Goal Structure

5 IMPLEMENTING B–BOUNDING

At a theoretical level, Behavioral Bounding, has very few requirements. All that is needed is a language capable of representing the agent's behavior and an ordering from specific to general over potential behavioral descriptions in this language. However, in order to make B–Bounding practical, a number of other constraints must be met. Most importantly, the language used to represent an agent's behavior must simultaneously be rich enough to distinguish between appropriate and inappropriate behavior, while also being constrained enough so that the learning problem remains tractable.

5.1 POTENTIAL REPRESENTATIONS

In order to determine a suitable language for describing agent behavior at a high–level, we should begin by examining the representation of a single observation of behavior. A single observation of behavior can be described by the list of ordered pairs $((S_1, B_1), (S_2, B_2), \cdots, (S_n, B_n))$ where ordered pair (S_i, B_i) indicates the behavior pursued in the given state. (Note that we can guarantee the uniqueness of the S_i by including the value of a world clock in the state description). At a minimum, each of the B_i encodes the external actions performed by the agent, but in some cases, B_i may contain additional information about the agent's behavior.

In the minimal case, where B_i contains only the externally observable action pursued by the agent, it is possible to abstractly represent the space of behaviors with canonical forms. Using this approach, states and actions are grouped into equivalence classes. For example, two distinct actions *Turn Right Quickly* and *Turn Right Slowly* might be grouped into a more generic *Turn Right* class. Similarly, particular features of the state space may be ignored within a portion of the problem domain yielding a set of equivalent states. For example, minor deviations from normal summer–time temperatures are unlikely to impact the manner in which an airplane is controlled.

When abstraction is used to reduce the complexity of the overall problem, each state and action in the observed behavior is replaced with the canonical form that represents the equivalence class of the observable. This representation has the ability to greatly reduce the size of the behavioral representation space, but this ability comes at a high cost. Using this method requires constructing these equivalence classes based on properties of the task that the agent is performing. Unfortunately, it is unlikely that an automated approach would be able to correctly identify these equivalence relations through direct examination of the environment. Instead, it is much more likely that this job would fall upon a human designer. Any additional human effort will increase the cost of performing the intelligence test, but constructing equivalence classes may be particularly costly because it must be done each time the test is performed in a new environment.

A less costly method of constructing a high–level representation of behavior can be performed if we assume the ability to gather more information about the agent's behavior. In particular, if we assume that the behavioral descriptions B_i contains both the agent's external actions as well as the agent's motivation for performing that action (i.e. the agent's current goals), we can use a high–level description of the agent's behavior without constructing a set of equivalence classes over states and actions.

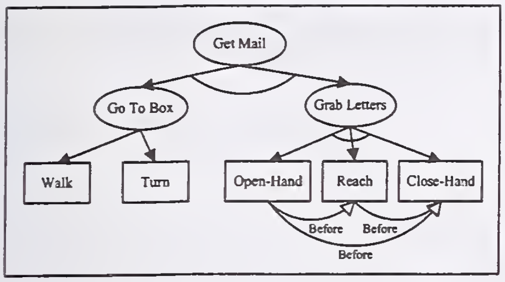


Figure 4: Constrained Goal Structure

5.2 EXPLOITING GOAL STRUCTURE

Because an agent's knowledge is structured around goals and actions, development of the agent's knowledge base is likely to profit from an explicit representation of these relationships. We will call this representation a goal hierarchy (see Figure 3) and it can be used to define a meaningful structure for the agent's knowledge similar to the way an outline serves to help structure the ideas in an article. The inner nodes in the goal hierarchy correspond to goals, while the leaves correspond to primitive actions. A node's descendants are the sub–goals and actions that may help to achieve the goal represented by the specified node.

The relationships described in the goal hierarchy make it possible to identify the set of primitive actions one would expect to observe from an agent pursing any particular goal. Thus, according to behavior represented by Figure 3, one would expect that an agent pursuing the goal of *Go To Box* would use the primitive actions *Walk* and *Turn*. A goal hierarchy in this form can be viewed as a general specification for behavior. By defining a set of constraints that can be added to the hierarchy, we can increase the specificity of the behavioral representation. This new language, formed by the goal hierarchy and constraints that can act upon it, can then be used by the B–Bounding technique to generate a concise representation of an agent's behavior.

To determine an appropriate set of constraints, we look to previous work in Hierarchical Task Networks (HTNs) [3, 1] from the planning community and Goal, Operator, Method and Selection–Rule (GOMS) models [2] from the HCI community.

HTNs consist of two types of nodes: goals and primitive actions. Methods indicate how goals decompose into sequences of sub–goals (or at the lowest level of abstraction, into primitive actions). Clearly, the HTN model has much in common with the goal hierarchy described above. However it also expands upon the basic structure in two ways:

1. HTNs explicitly represent alternative ways of decomposing a goal into sub–goals.
2. HTNs are able to explicitly represent ordering constraints between siblings nodes (so, for example, a method can describe a specific order in which the sub–goals must be accomplished).

GOMS models also consist of two types of nodes: goals and operators, which map directly onto the primitive actions of HTNs. Methods in a GOMS model performs the same function as in HTNs, and it is assumed that these methods have the ability to represent ordering constraints on the nodes in the decomposition. The main difference between HTNs and GOMS models is that the later use explicit structures (selection–rules) to determine the set of methods that are appropriate for a particular situation.

Our representation language is constructed from the common components that underlie both HTNs and GOMS models. Specifically, we have two distinct types of constraints: goal types and ordering constraints. Goals can be either of two types AND or OR. An AND node indicates that in order to accomplish the goal represented by the node, all of the children must be accomplished. An OR node indicates that one (or more, but not all) of the children must be accomplished in order to accomplish the goal. By using multiple levels of the hierarchy AND and OR nodes can be used to represent alternative *methods* for solving a higher–level goal, just as in HTNs and GOMS models. The second type of constraint in our model indicates the valid orders in which goals or actions can be accomplished. For this purpose, we use binary temporal constraints, to build a partial ordering between the children of each goal node. Figure 4 illustrates a goal hierarchy that is partially constrained. In this example, the constraints indicate:

- Achieving the goal *Get Mail* requires accomplishing both of the sub–goals *Go To Box* and *Grab Letters*.
- Achieving the goal *Grab Letters* requires performing the actions *Open Hand*, *Reach*, and *Close Hand* in that order.

With these two types of constraints we can specify a general to specific ordering over abstract behavior representations. In this ordering, the maximally general behavior specification is one in which all nodes are of type OR, and no temporal constraints exist. The maximally specific behavior representation in which all goals are of type AND and there is a total ordering between all siblings. Because these constraints underlie both GOMS models and HTNs, we can be confident that they will provide a good medium for representing behavior in a variety of domains. Furthermore, because the goal–hierarchy itself can be used as an organizational tool to help outline and develop the agent's knowledge, it is likely that this representation can be used with little cost.

6 CONCLUSION

The definitions put forth in Newell's book *Unified Theories of Cognition* provide a basis for constructing a simple algorithm to measure the intelligence of a particular system. However, a straightforward implementation of this metric may require impractical computational resources when a measurement of general intelligence is required. We have presented the Behavioral Bounding method that mitigates this problem by ensuring that the minimal amount of computation is performed to gain a measurement of an agent's intelligence. In addition, Behavioral Bounding can be used to explicitly indicate the relative generality of a particular intelligence measurement, thus overcoming another flaw in the original algorithm without incurring additional cost.

Recently, our research has been focusing on using the Behavior Bounding method to validate an agent's knowledge based on observations of its behavior (the third application of B–Bounding outlined in Section 2). We have implemented and begun an investigation of Behavioral Bounding using the goal hierarchy and constraint language discussed in Section 5.2. Future work will focus on enriching the constraint language and new uses for this technique.

REFERENCES

[1] Kutluhan Erol, James Hendler, and Dana S. Nau. UMCP: A sound and complete procedure for hierarchical task–network planning. In *Artificial Intelligence Planning Systems*, pages 249–254, 1994.

[2] Bonnie E. John and David E. Kieras. The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer–Human Interaction*, 3(4):320–351, 1996.

[3] Subbarao Kambhampati. A comparative analysis of partial order planning and task reduction planning. *SIGART Bulletin*, 6(1):16–25, 1995.

[4] Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.

[5] Allen Newell. The knowledge level. *AI Magazine*, 2(2), 1981.

[6] Allen Newell. *Unified Theories of Cognition.* Harvard University Press, Cambridge, Mass, 1990.

PART I
RESEARCH PAPERS

2P2 - Modeling of Mind

1. Beyond the Turing Test: Performance Metrics for Evaluating a Computer Simulation of the Human Mind
   N. Alvarado, IBM Corporation
   S. Adams, IBM Corporation
   S. Burbeck, IBM Corporation
   C. Latta, IBM Corporation

2. Experimental Evaluation of Subject Matter Expert-oriented Knowledge Base Authoring Tools
   R. Schrag, Information Extraction & Transport (IET), Inc.
   M. Pool, Information Extraction & Transport (IET), Inc.
   V. Chaudhri, SRI International
   R. Kahlert, Cycorp, Inc.
   J. Powers, Information Extraction & Transport (IET), Inc.
   P. Cohen, University of Massachusetts
   J. Fitzgerald, Information Extraction & Transport (IET), Inc.
   S. Mishra, University of Massachusetts

3. A Task Domain for Combining and Evaluating Robotics and Cognitive Modeling Techniques
   N. Cassimatis, Naval Research Laboratory
   G. Trafton, Naval Research Laboratory
   A. Schultz, Naval Research Laboratory
   M. Bugajska, Naval Research Laboratory
   W. Adams, Naval Research Laboratory

4. Modelling, Self and Consciousness: Further Perspectives of AI Research
   R. Sanz, Universidad Politecnicia de Madrid
   A. Meystel, Drexel University

Beyond the Turing Test: Performance Metrics for Evaluating a Computer Simulation of the Human Mind

Nancy Alvarado, Sam S. Adams, Steve Burbeck, Craig Latta
IBM, Thomas J. Watson Research Center
{nanalv, ssadams, sburbeck, crl}@us.ibm.com

Abstract

Performance metrics for machine intelligence (e.g., the Turing test) have traditionally consisted of pass/fail tests. Because the tests devised by psychologists have been aimed at revealing unobservable processes of human cognition, they are similarly capable of revealing how a computer accomplishes a task, not simply its success or failure. Here we propose the adaptation of a set of tests of abilities previously measured in humans to be used as a benchmark for simulation of human cognition. Our premise is that if a machine cannot pass these tests, it is unlikely to be able to engage in the more complex cognition routinely exhibited by animals and humans. If it cannot pass these sorts of tests, it will lack fundamental capabilities underlying such performance.

1. Introduction

What constitutes success for designers using computational autonomous development to create intelligent systems? The observation of change over time with experience in an environment is neither learning nor development unless that change results in closer approximation to some performance goal. Further, neither development nor learning necessarily indicates that a machine has behaved intelligently. While it may be interesting to observe how systems respond to different environmental challenges and how parameters affect a system's behavior, eventually some attempt to guide learning must be made if these systems are to shed much light on human cognitive processes or prove useful in applications.

This was the challenge we faced in developing a set of performance goals for the Joshua Blue Project. Joshua Blue applies ideas from complexity theory and evolutionary computational design to the simulation of a human mind on a computer. The goal is to enhance artificial intelligence by enabling the emergence of such capacities as common sense reasoning, natural language understanding, and emotional intelligence, acquired in the same manner as humans acquire them, through situated learning in a rich environment. Because our goal is to simulate human cognition, our performance goals arise from observing how humans behave in similar circumstances. Thus our project has been guided by findings in developmental and cognitive psychology.

If a system is conceived as a "black box," engineers work from inside the box to implement the capabilities that will achieve a desired outcome within a given set of constraints. In contrast, psychologists work from outside the box, reverse-engineering what already exists in the human mind to develop plausible explanations for observed behavior. These different interests converge in the need to specify performance goals for autonomous learning systems because, as in psychology, such testing must be performed from outside the box. Thus, the paradigms used by psychologists to test hypotheses about how the human mind functions provide a ready metric for assessing how closely computer behavior approximates human performance in well-defined contexts. In these tests, the observed behavior of an autonomous system becomes proof of what must be occurring inside a computer black box designed by an engineer. This is important for those systems that cannot be inspected directly – where the changes taking place are hidden from the system's designers or difficult to interpret, as is typically the case with architectures based on artificial neural networks.

For many purposes, it may not matter that a computer accomplishes a task in a manner similar to humans. In our case, because the goal of the Joshua Blue Project was to simulate human understanding, thinking in the same manner as humans has been an important design goal. Our reasoning has been that every human capacity exists within the interdependent context of the human mind for a specific reason. Thus, approximating human functioning as closely as possible in all respects seems necessary to success at modeling a mind with rich, robust human-like cognitive abilities. This goal is not as restrictive as it seems, given the variation that exists among humans. Experience may teach us which abilities are essential, which are not, and how much deviation is possible without sacrificing the cognitive qualities of the resulting system.

When a comparison between machine and human functioning is proposed, it seems natural to ask "which human shall we compare our machine with?" This question arises because humans do not all perform alike. Psychologists have addressed this by describing behavior in terms of distributions. The "normal" or bell curve for intelligence is one such description of the range of human performance on a specific measurement instrument (such as an IQ test). Such curves permit one to place a particular score or measured behavior within the context of behavior for a large number of others. Thus it provides not simply pass/fail information but a basis for comparison with the range of human variation. Norms exist for most published tests. When machines become more autonomous in their learning and have greater flexibility in what they can do, when their learning experiences are richer and more varied, governed by the machine's own choices, the behavior of intelligent systems will also be less predictable and more variable. Statistical techniques for describing variance and comparing behavior, like those used by psychologists, will become more important in assessing machine performance. Such techniques will be needed to determine whether different observed behavior reflects learning or simply change.

One longstanding test of human-like computer intelligence has been the Turing Test [1], now conducted each year as the Loebner Competition [2]. A drawback of that test is that human functioning can be mimicked by systems that have little if any human-like cognitive capabilities. The performance goal can be met by simulating human behavior convincingly (to the satisfaction of human judges) for a relatively brief period of time in a restricted domain of discourse. Surface behavior is partially simulated but the underlying cognition producing that behavior is not. We believe that meaningful performance tests must incorporate the idea that it matters how behavior is accomplished, not simply that it occurs [3].

When a developmental process is considered important to acquiring human-like understanding, as we believe it to be, then a dimension of change in stages with milestones approximating human development is introduced. The time frame for a computer need not be the same as for a human child because its training experiences may differ, but the order of acquisition of abilities, the interrelationship of change in one domain with change in another, and so on, should be considered. Taking a modular approach, as Fodor suggests, temporarily simplifies evaluation of developmental sequence and inter-relationships among processes, but at some point separate processes become unified, domain-specific knowledge converges, sensory data is integrated, and more general or encompassing cognition emerges.

Ultimately, as a measure of success, we envision Joshua Blue acquiring the capacity to pass not an adult Turing Test, but a "Toddler Turing Test." Success would mean that our system would answer unrestricted questions posed by adults in the same manner as a three year old toddler would, with all of the common sense, naïve physics, burgeoning language use, and social understanding of a preschool child. The true test would be whether Joshua Blue is distinguishable from human three year olds, not whether Joshua Blue fools an adult into believing it is not a computer, through faked typing errors, colloquialisms, illogic or scripted emotional responses.

2. Overview of the Test Suite

The proposed test suite is intended to assess whether certain essential, prerequisite cognitive abilities exist in a system. This required identification of foundational abilities without which a system cannot be considered a simulation of human cognition. The approach to testing for these abilities is twofold. First, changes in internal system values and other parameters indicating changes in mental state can be monitored and measured, and related to observed behavior. Second, well-understood paradigms from psychology can be adapted for use as performance benchmarks. In many cases, such paradigms have demonstrated existence of unobservable cognitive processes through observable and quantifiable behavior. By comparing system results to results obtained with animals and humans, designers can make stronger claims about the capabilities of that system.

The test suite is divided into three parts, as shown in Table 1. The goal for systems simulating the general properties of human cognition must be to pass all of the tests using a single, generalized system. Ultimately, passing an unrestricted toddler Turing test would require success in all three parts of this initial test suite, plus a great deal more.

3. Tests of Associative Learning

We believe that a system capable of learning from its environment must be endowed with motivation that gives it the impetus to explore its world, recognize reward and punishment, form goals and seek to satisfy them. While certain drive states may be innate to the system, it should also form acquired goals and use past experience to determine a course of action. This test suite is designed to assess a system's ability to direct its own behavior by recognizing environmental cues that signal reward or punishment. The tests are based upon animal testing in the field of behavior modification and learning. An excellent overview of this work is presented by Klein [4].

3.1 Tests Showing Formation of Associations

In classical conditioning, associations are formed between environmental cues (called conditioned stimuli) and elicitors of affective or motivational states (called unconditioned stimuli). Training occurs by presenting these two kinds of stimuli together until one becomes a predictor of the other and is responded to as if it were the other stimulus. Passing this test requires the ability to recognize salient features of the environment and associate them with the internal responses evoked by a co-occurring unconditioned stimulus.

Table 1. Overview of Performance Metrics for Evaluating a Simulation of Mind

Associative Learning	Classical Conditioning	Forms associations between a predictive environmental cue and an accompanying stimulus
	Presence of a Mind	Forms mental representation of two cues as a single predictor, demonstrates memory for previously encountered predictive cues
	Instrumental Conditioning	Demonstrates cause-effect learning and responds to changes in amount or rate of reward or punishment
	Purposeful Behavior	Acquires goals and expectations, shows "learned helplessness" with a lack of ability to achieve goals, shows escape/avoidance behaviors
Social Cognition	Social Encoding	Categorizes self and others by salient features and behavior, forms stereotypes, forms a self schema
	Social Inference	Forms expectations (heuristics) based on observed correlations and covariation among properties and behaviors of others, exhibits biases resulting from use of such heuristics
	Causal Attribution	Attributes motives to others, exhibits biases resulting from such attributions, generalizes its self schema to others
	Representation of Self	Forms a representation of self observable in self-preserving or enhancing behavioral choices in contexts that threaten self-image
	Empathy and Attachment	Generalizes the sense of self to others, demonstrates changes in affect depending upon affiliation with others, shows affect-guided helping behavior
Language Acquisition	Prelinguistic Structural Competences	Performs combinative operations on sets, acquires phoneme combinations appropriate to a specific language, progresses through cooing to babbling and prosody mimicking speech, acquires pragmatics of communication, understands communicative intentions of others
	Intentional Communication	Acquires and uses words to accomplish goals both through associative learning and imitation of observed others
	Word Acquisition	Sorts words by grammatical types (e.g., noun, verb), learns grammatical rules in same sequence as children, overgeneralizes grammatical rules
	Cross-Language Comparisons	Demonstrates language-specific competence in a language other than the first language taught, repeating the three tests above.

Given the capacity to form associations through experience in an environment, classical conditioning can be tested in a virtual environment using an embodied agent endowed with minimal sensors and effectors, motives, affect, and proprioception of both actions and affective states. In order to pass the classical conditioning tests, the system must form a classically conditioned association between an unconditioned stimulus (e.g., satisfaction of "hunger") and a conditioned stimulus (cues signaling availability of "food"). This association should result in observed changes in behavior. If a pleasant stimulus is replaced by an aversive (unpleasant) stimulus, then fear-based conditioning should result in avoidance and escape behaviors.

A rich literature described in most learning theory textbooks [4, 5] specifies quantitative relationships between factors affecting learning and observed behavior. These experiments can be used as a highly specific model against which the behavior of a system can be compared.

3.2 Tests Demonstrating Mental Representation

A series of studies in the learning literature were used to demonstrate that certain classical conditioning effects are the result of expectations, which cannot exist unless experience is being represented mentally. Several phenomena can be used to show the existence of mental representations in a system. This use of behavioral paradigms to demonstrate existence of mental representations illustrates that the goal of a study that measures behavior can be far more than simply to demonstrate competence in performing that behavior. It can also demonstrate indirectly what must exist in the less observable parts of the system. The reasoning is that the observed behavior, although often uninteresting in its own right, could not be performed if certain important processing were not occurring. Two examples are presented here, but there are many more in the learning literature.

Normally, when two cues to an event are present, the more salient cue (the better predictor) will block conditioning of the lesser cue. However, if the two cues are perceived as part of a unified percept, instead of as two separate events, then the opposite effect occurs and presence of the highly salient cue potentiates the response to the less noticeable cue. This potentiation cannot occur without formation of a mentally unified percept incorporating both cues. Similarly, in backward blocking, a new aversive event becomes associated with a previous environmental cue not actually present during the unpleasant experience. If no memory of the cue were formed, it could not become associated with the aversive stimulus that came later. This provides an indirect way of confirming that a system is forming mental representations of its world and associating percepts with each other

3.3 Cause-Effect Learning Tests

Operant and instrumental conditioning (cause-effect learning) paradigms test whether associations are formed between environmental stimuli and voluntary behavior, governed by reward or punishment. Virtual or physical environments similar to the mazes and runways presented in classic psychology experiments can be created to test such learning. These include runways along which an agent or robot can move, T-mazes and radial-arm mazes permitting choice of paths, two-compartment shuttle boxes to test escape and avoidance learning, more complex mazes like those used by Tolman [6] to demonstrate existence of cognitive maps, virtual or physical "Skinner boxes" (single and two-choice lever-pressing boxes). Equivalent behaviors can be devised, such as moving to a specific region of the environment to obtain reward.

To pass these tests, a system must show a reliable relationship between reward, punishment, and subsequent behavior. The system should increase its behavior in response to a reinforcer, decrease it in response to a punishment or in the absence of a reinforcer. It should also conform to effects of changes in size, rates and timing of reinforcement, as described in the literature on schedules of reinforcement [4, 5]. Further tests of complex contingencies can show that the system has the capacity to monitor longer chains of behavior and form more complex goals.

Learning theory includes descriptions of several more sophisticated learning phenomena involving expectations, multiple reinforcers and behavioral allocation (dividing attention between two sources of reward). We expect that a system should show similar behaviors, including Crespi's depression effect, punishment effects, a partial reinforcement effect, adherence to Premack's principle and Herrnstein's matching law, all classic learning phenomena described in textbooks [4, 5]. Because stimulus control phenomena are closely related to ability to discriminate, identify relevant features, and form concepts and categories, they are not discussed here, but are described in Section 5.

3.4 Tests of Purposeful Behavior

Simple cause-effect learning does not necessarily imply that an organism has formed a mental representation of its goal states. Tolman's [6] paradigms can be adapted to demonstrate existence of purpose, expectations, and mental maps in a system. A system must be capable of representing aspects of its environment for future use, even when those representations are not directly related to a current operation or goal state. Latent learning studies can be used to show that mental representations are being formed without reward.

Learned helplessness and phobias are both examples of phenomena demonstrating the existence of expectations. Learned helplessness occurs when an organism is subjected to punishment but can do nothing to escape it, and it results in a change in expectations about effectiveness of future activity. Phobias represent acquisition of a classically conditioned association between some environmental cue and an aversive event. Observation of phobias indicates formation of expectations about negative events, plus beliefs about what behaviors will prevent those events. Suitable paradigms for testing each of these can be adapted.

4. Tests of Social Cognition

Tests of two or more agents interacting in a shared environment permit application of a system's basic cognitive capacities to social interactions. In addition to the capacities needed to function alone in such an environment, the system must communicate, compete and form alliances with other agents, understand the behavior and intentions of others, and find ways to coexist with them while also attaining important goals. Fiske and Taylor [7] provide a thorough overview of paradigms for studying human social cognition.

4.1 Tests of Social Encoding Processes

Social encoding can be tested in a simplified, virtual multiple-agent environment by creating two types of embodied agents: (1) agents with similar bodies and behavior; (2) agents with dissimilar bodies and behavior. Scenarios can be manipulated so that these agents compete with, punish (show aggression), or cooperate with and help the system being tested. From this experience, the system should form an appropriate mental representation of both like and unlike agents and behave according to its previous experience with those agents. Stereotypes should be formed as mental representations of the characteristics of these groups of agents. Stereotypes can be demonstrated by observing an experienced system's behavior toward an ambiguous new agent that includes some but not all of the salient characteristics of previously learned social categories. Once social categories have been learned, increased ambiguity in classification of new agents should result in interruptions of behavior and affective distress in the system.

4.2 Tests of Social Inference

Social inferences are based on observed correlations and covariation among properties and behaviors of other agents and environmental events and features. Once learned, these become heuristics used in human reasoning. Existence of such heuristics is observed in specially devised contexts where they produce inference "errors." If a system has acquired heuristics, then it should demonstrate the same errors as humans, instead of strictly mathematical/statistical decision processes. These should result in the biases enumerated in most textbooks [7], including: (1) framing effects, use of extreme cases, over-reliance on small samples or biased samples; (2) salience of immediate experience; (3) inability to combine joint probabilities; (4) inability to identify diagnostic information; (5) inability to correct for regression artifact; (6) irregular or improper weighting of cues.

4.3 Tests of Causal Attribution

Humans recognize that living beings have agency – the ability to control their own actions in goal-directed ways. Thus they attribute motives to others and understand the actions of others in terms of intentions. While people understand that circumstances can dictate behavior, they prefer to attribute the actions of others to inherent motives, preserving a belief in free will. This is reflected in certain attribution biases [7], including the fundamental attribution error, the actor-observer effect, and self-serving biases. Environments can be created for testing fundamental attribution error, the false consensus effect (where we consider our behavior to be more representative of the behavior of others than it really is), and self-serving bias.

4.4 Tests of a Representation of the Self

People tend to form representations of the self (self schemas) along dimensions that are important to them in their lives. Complexity of the self-schema is also determined by life experiences. Thus a system's self-schema should depend upon its experiences. In humans, tasks involving perception, memory and inference are used to demonstrate the characteristics of the self schema. These can be adapted to show that a system has formed a schema whose characteristics vary with experience [7]. For example, self-schemas change to more closely resemble those of others surrounding a person. Thus the existence of a self-schema can be demonstrated through changes in behavior that occur when the system is placed in a new environment, among different social agents than were present when the self-schema was formed. Maintenance or preservation of a self-schema motivates much human behavior. A cybernetic theory of self-attention and self-regulation has been proposed by Carver & Scheier [8]. This theory of regulation, with predicted consequences for failure, provides a detailed model for testing a system's self-schema in different contexts.

4.5 Tests of Empathy and Attachment

In humans, empathy and attachment are the two main mechanisms for regulating social interactions. Both depend upon the cognitive evaluations and interpretations made of social situations. When a system is endowed with affective responses, these should mediate social interactions in ways consistent with what is observed in humans. Tests of emotion-motivated helping behavior can be adapted to assess empathy [7]. Berscheid [9] proposes a theory that predicts how emotion changes in long-term relationships, readily testable in the simple social contexts described earlier.

5. Tests of Language Acquisition

The criteria for assessing machine language acquisition are that a computer's language must be used with the flexibility of a real language, but must not be so constrained that one language can be learned but not another (e.g., English but not Chinese). Two aspects of language acquisition are intertwined: (1) understanding of the structural properties of the representations comprising language; (2) understanding of the content of concepts represented by language. Understanding of concepts appears to precede language acquisition, although a child's preexisting concepts are in turn modified by later language learning. The complexities of the structure of language are so great that many theorists hypothesize innate mechanisms for acquiring structural understanding, and even for acquiring conceptual understanding.

An important prerequisite to language learning is the ability to perform combinative operations on sets. Langer [10] provides convincing evidence that this ability is needed in order to form the class inclusion hierarchies essential to categorization. He also argues that these operations (recursive mapping of cognitions, correspondence mapping, substitutions and equivalence relations) are the foundation of abstraction and form the basis for linguistic rewrite rules.

Some theorists, such as Pinker, hypothesize a straightforward mapping of words to "mentalese" nonverbal representations, guided by innate mechanisms. Other theorists assert that general associationist learning can account for language acquisition, especially if social imitative learning directs a child's attention to what is relevant during learning [11, 12]. Thus, a system must acquire language socially and must have the mechanisms for imitative social learning in place before acquiring language.

5.1 Tests of Prelinguistic Structural Competences

The first year of life is spent acquiring prelinguistic competences supportive of ultimate language development. These include a progression from cooing and laughing to vocal play, babbling, and utterances that mimic the prosody of speech, acquisition of the nonverbal pragmatics of communicative interactions, such as smiling, gaze orienting, turn-taking, pointing and showing, and mimicking behaviors. The child also acquires an understanding of the communicative intentions of adults [12]. A first task is to learn to segment a flow of speech sounds into meaningful combinations of phonemes (morphemes). The system's developing language should show increasing inclusion of language-specific permissible combinations of phonemes. A set of tests requiring the system to classify items into sets and to perform the combinative operations on those sets described by Langer [10] can assess whether the conceptual cognitive abilities prerequisite to language learning exist.

5.2 Tests of Intentional Communication

Intentional communication requires use of language instrumentally to accomplish goals and an understanding of the ways in which others do so. The social paradigms described earlier can be adapted to create situations in which a system must learn to use words to accomplish specific goals. Placed into a new environment, the system must be able to learn to use words through observation and imitation of the behavior of other social agents inhabiting the same environment.

5.3 Tests of Word Acquisition

An understanding of grammar is necessary for word acquisition because grammar constrains the possible meanings of new words. While this grammatical knowledge may be innate, some theorists argue that the statistical frequencies in language call attention to the structural properties of language sufficiently to permit grammatical knowledge to emerge [11]. Either way, tests of word acquisition begin with tests of grammatical knowledge. A system must be able to tell the difference between nouns, verbs, and adjectives, in a simple sorting task. Further, the system must use that knowledge to classify objects. This can be tested by adapting paradigms used with young children, such as those demonstrating shape bias with nouns [11]. Tests of more sophisticated grammatical knowledge demonstrate rule acquisition by observing occurrence of errors resulting from the overgeneralization of rules. These errors should gradually disappear with greater experience.

5.4 Cross-Language Comparisons

Once a system has demonstrated competence in one language, it must pass the same set of tests to demonstrate competence in a different language.

6. Conclusion

There are many other aspects of cognitive functioning that might be assessed. We propose this battery as a place to start. If a system cannot pass these tests, it is unlikely to be able to engage in the more complex cognition routinely exhibited by animals and humans. Because all of these tests focus upon abilities quantitatively measured in humans, they provide a ready-made benchmark for simulation of human cognition. The greater challenge is to create learning environments sufficient to enable machines to acquire these abilities if they are truly capable of doing so.

Notes

This paper was originally presented at the Second International Conference on Development and Learning, ICDL 2002, MIT, Cambridge, MA. Portions have been revised in response to comments by conference attendees. Nancy Alvarado may be reached at the University of California, San Diego, Center for Brain and Cognition (alvarado@psy.ucsd.edu).

References

[1] A. M. Turing, "Computing Machinery and Intelligence," In *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, A. Collins & E. Smith (Eds.), pp. 6-19, Morgan Kaufmann, San Francisco, CA, 1950/1991.

[2] H. Loebner, "Loebner Prize," described at: http://www.loebner.net/Prizef/loebner-prize.html.

Breazeal, C. 2001. *Designing Sociable Machines*. The MIT Press. Forthcoming.

[3] J. R. Searle, 1980. Minds, brains, and programs, *Behavioral and Brain Sciences, 3*, 417-424.

[4] S. B. Klein, 1996. *Learning: Principles and Applications*. McGraw-Hill Inc., New York, NY.

[5] G. Bower & E. Hilgard, 1981. *Theories of Learning*. Prentice-Hall, Inc., Englewood Cliffs, NJ.

[6] E. C. Tolman and C. H. Honzik, 1930. *"Insight" in Rats.* University of California Publications in Psychology, 4, 215-232.
[7] S. T. Fiske and S. E. Taylor, 1991. *Social Cognition.* McGraw-Hill Inc., New York, NY.
[8] C. S. Carver & M. F. Scheier, 1981. *Attention and Self-regulation: A Control Theory Approach to Human Behavior*, Springer-Verlag, NY.
[9] E. Berscheid, M. Snyder, & A. Omoto, 1989. Issues in studying close relationships: Conceptualizing and measuring closeness. In C. Hendrick (Ed.), *Review of Personality and Social Psychology: Vol 10. Close Relationships* (pp. 63-91). Sage, Newbury Park, CA.
[10] J. Langer, 2001. The mosaic evolution of cognitive and linguistic ontogeny. In M. Bowerman and S. Levinson (Eds.), *Language Acquisition and Conceptual Development.* Cambridge University Press, New York, NY.
[11] L. Smith, 2001. How domain-general processes may create domain-specific biases. In M. Bowerman and S. Levinson (Eds.), *Language Acquisition and Conceptual Development.* Cambridge University Press, New York, NY.
[12] M. Tomasello, 2001. Perceiving intentions and learning words in the second year of life. In M. Bowerman and S. Levinson (Eds.), *Language Acquisition and Conceptual Development.* Cambridge University Press, New York, NY.

Experimental Evaluation of Subject Matter Expert-oriented Knowledge Base Authoring Tools

Robert Schrag[1], Mike Pool[1], Vinay Chaudhri[2], Robert C. Kahlert[3], Joshua Powers[1], Paul Cohen[4], Julie Fitzgerald[1], and Sunil Mishra[4]

1) Information Extraction & Transport (IET), Inc.
1911 North Fort Myer Drive #600
Arlington, Virginia 22209
last-name@iet.com

2) SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
{vinay, smishra}@ai.sri.com

3) Cycorp, Inc.
3721 Executive Center Drive #100
Austin, Texas 78731
rck@cyc.com

4) Department of Computer Science
University of Massachusetts
Amherst, MA 01003
cohen@cs.umass.edu

ABSTRACT

We describe a large-scale experiment in which non-artificial intelligence subject matter experts (SMEs)—with neither artificial intelligence background nor extensive training in the task—author knowledge bases (KBs) following a challenge problem specification with a strong question-answering component. As a reference for comparison, professional knowledge engineers (KEs) author KBs following the same specification. This paper concentrates on the design of the experiment and its results—the evaluation of SME- and KE-authored KBs and SME-oriented authoring tools.

Evaluation is in terms of quantitative subjective (functional performance) metrics and objective (knowledge reuse) metrics that we define and apply, as well as in terms of subjective qualitative assessment using several sources. While all evaluation styles are useful individually and exhibit collective power, we find that subjective qualitative evaluation affords us insights of greatest leverage for future system/process design. One practical conclusion is that large-scale KB development may best be supported by "mixed-skills" teams of SMEs and KEs collaborating synergistically, rather than by SMEs forced to work alone.

KEYWORDS: *knowledge acquisition, evaluation*

1 Introduction

The authors are engaged in a joint research program—Rapid Knowledge Formation (RKF)—to develop and evaluate technology to enable SMEs to build very large KBs. Two teams respond to challenge problems posed by an independent evaluator. We report on a large-scale evaluation conducted during the summer of 2001.

The RKF teams are led by Cycorp and SRI International. The independent evaluator is IET. More comprehensive information about the evaluation—including a full challenge problem specification—is available at http://www.iet.com/Projects/RKF/. For more general program information, see http://reliant.teknowledge.com/RKF/.

In the remainder of this paper, we first outline our approach to evaluating KBs. Then we describe the "textbook knowledge" challenge problem (TKCP) presented to SMEs and KEs for KB authoring, teams' tools, experimental procedures, and results from each style of evaluation. We close with discussion/conclusions.

2 KB Evaluation Approach

We consider KB evaluation along three dimensions: functional performance (with subjective metrics), economics (with objective metrics), and intrinsic quality (subjective and non-metric). Here we elaborate on these dimensions and our methodology. In a later section we describe results.

To evaluate functional performance, we follow Cohen et al. [3] in posing test questions (TQs) to authored KBs and scoring their answers against defined criteria. Our criteria fall into three major categories: Representation (with criteria Query Formulation, Term Quality, and Compositionality), Answer (with criterion Correctness, only), and Explanation (with criteria Content Adequacy, Content Relevance, Intelligibility, and Organization). While the Answer category obviously addresses a KB's functional performance, we argue that high-quality question representations and explanations also confer valuable (input- and output-oriented) functionality to KBs.

To evaluate economics, we follow Cohen et al. [2] in addressing reuse—the extent to which knowledge created earlier is exploited in the creation of subsequent knowledge. We require that authored knowledge (including constants and axioms) bear labels of authorship and creation time. Other things being equal, greater reuse is considered more economical.

Others—[4], [5]—have suggested (without employing, to our knowledge, in large-scale comparative evaluation) qualitative criteria for assessing intrinsic properties of KBs and ontologies. Inspired by these, we formed a KB Quality Review Panel from among technology providers and evaluators to assess the following properties: Clarity or Style, Maintainability or Reusability, Correctness or Accuracy, Appropriate Generality, Appropriate Organization, and Logical Propriety. While we discussed making this evaluation quantitative (by adapting our Functional Performance scoring methodology described below), the Panel ultimately agreed that free-form commenting along these dimensions would be the most fruitful initial step.

We drew on two other sources, besides the Panel, in our subjective qualitative evaluation: post-evaluation SME survey responses and evaluator observations. Findings regarding RKF tools' strengths and weaknesses were consistent across all three sources. RKF tool developers have taken these results seriously and have begun appropriate modifications to their tools.

2.1 Additional Related Work

The series of Knowledge Acquisition Workshops (KAWs)[1] has emphasized the evaluation of generic problem-solving methods (PSMs) and performance on the associated problem-solving tasks more than that of knowledge for its own sake. This appears to reflect a difference in emphasis or philosophy: whereas the KAW community has focused on the PSM as the primary reusable artifact, the RKF community has focused on KBs themselves as reusable artifacts that should, in principle, be applicable to any problem-solving task.

3 Textbook Knowledge Challenge Problem

The TKCP's KB authoring task is to:

1. Capture knowledge about DNA transcription and translation from about ten pages of an introductory undergraduate molecular biology textbook for non-majors [1];
2. Ensure that the authored KBs are capable of correctly answering test questions about the subject material, (extending or revising KBs as necessary).

We chose a textbook source because it serves as a circumscribed reference that offers an intuitively justified basis for required KB content scope. We chose molecular biology because it is a largely descriptive science and because it is of interest to the sponsor. We chose [1] because it largely eschews description of laboratory procedures or scientific history in favor of material phenomena.

The TQs were consistent in difficulty with TQs typically found on Web-available quizzes on molecular biology. Questions appearing in the textbook itself typically required representation of (e.g., hypothetical/counter-factual) situations that were entirely novel compared to the basic material presented in the text. These were judged by the RKF community to be unsuitable (too difficult) for use in evaluation of current SME-oriented KB authoring technology. The TQs were similar in style and difficulty to IET-created sample questions (SQs) covering material in earlier chapters of the textbook. SQs were provided to teams before the evaluation. TQs were not so disclosed.

Besides the primary KB authoring tools described in the following section, RKF teams were required to include facilities for SMEs to pose TQs and to package their answers for evaluation. They also were required to prepare various instrumentation capabilities in support of metrics computations.

Teams' tools included substantial TKCP-relevant knowledge before they were handed off to SMEs. Given the premise that a large, general/reusable KB facilitates the construction of more specific KBs, teams were allowed to "prime the pump" of knowledge development by seeding KBs with prerequisite (e.g., pertaining to earlier—largely review—textbook chapters) and background (including high-level/abstract) knowledge or reasoning abilities deemed appropriate (according to defined ground rules) to support the authoring of the textbook's target knowledge.

4 Tools under Evaluation[2]

Cycorp's "KRAKEN" tools are supported by a substantial KB based on a higher-order formal predicate logic. The key strategies of SME-oriented KB interaction are natural language (NL) presentation and a knowledge-driven acquisition dialog with limited NL understanding. The KB includes thousands of predicates and understands thousands of English verbs. Cycorp's approach might be described as maximalistic, domain-pluralistic, and conceptually precise. The KRAKEN tools aim to exploit (as leverage) a substantial KB to bring SMEs past an otherwise-steep learning curve by productive collaboration in this sophisticated knowledge representation milieu.

SRI's "SHAKEN" tools are supported by a relatively sparse KB based on the frame formalism. The key strategy of SME-oriented interaction is graphical assembly of components. The KB includes a few hundred predicates serving as conceptual primitives (the components). SRI's approach might be described as minimalistic, domain-universal, and conceptually coarse. The SHAKEN tools may be seen as skirting traditional knowledge representation complexity by presenting an entirely new metaphor with great intuitive appeal.

5 Experimental Procedures

IET collaborated with George Mason University (GMU) to establish a SME KB authoring laboratory at GMU's Prince William County, Virginia campus. Eight (mostly graduate) biology students participated in the TKCP evaluation, four

[1] See http://ksi.cpsc.ucalgary.ca/KAW/.

[2] More detailed tool descriptions appear in appendices. Here we include the briefest salient sketches.

working with Cycorp's KRAKEN tools, four with SRI's SHAKEN tools. All worked full-time from mid-May until mid-July, 2001. The first week of this period was devoted to classroom-style training of SMEs by teams. The next two weeks were taken up with an evaluation dry run that included shake-down of tools in the installed context and limited additional, informal training. The evaluation-proper was held during the TKCP's final four weeks. It covered about seven pages of the textbook and included 70 TQs (about 3 pages and 10 TQs having been covered in the dry run). The actual test material covered five subsections of the textbook's target material. SMEs were allowed to author this material in any order they liked, but IET would not release one subsection's TQs to a SME until s/he had completed work on TQs for earlier subsections.

Subsequent to training, SMEs had no direct contact with the teams' KEs. Instead, to deal with tool understanding issues that might arise, IET staffed the SME lab full-time with a "gatekeeper" KE who mediated contacts with the teams (including bug reports and fixes). The gatekeeper KE also provided a subjective window on SME activity. Teams were allowed to augment KBs during the evaluation in accordance with the TKCP's pump priming ground rules.

Besides these SMEs, two KEs from each team also participated (off-site from the SME lab) by addressing the same KB authoring tasks using tools of their choice. SRI KEs used the same SHAKEN tools available to the SRI-assigned SMEs. Cycorp KEs usually authored knowledge in CycL (a KE-oriented knowledge representation language) using a text editor, rather than with the SME-oriented tools in KRAKEN. Cycorp KEs did not author all target textbook knowledge during the evaluation. Instead, they relied on a base of target knowledge that Cycorp had first developed in support of its internal pump priming requirements identification, then excised before tool delivery to SMEs. (This was due to unavoidable personnel overlap between Cycorp's pump-priming and TKCP-participating KEs.) SRI KEs were given the same option but elected to author the textbook knowledge during the evaluation. All KEs authored TQ representations and developed answers independently.

SMEs and KEs participants were required to answer at least 75% of the TQs presented for each subsection. In the results below, we include for each subsection the 75% of each participant's answered questions with the highest overall scores, padding with 0s as necessary. One of these subsections ("Signals in DNA Tell RNA Polymerase Where to Start and Finish") was particularly troublesome for the Cycorp SMEs. After they had spent well over a week working on it and were all well less than halfway to reaching their answered-TQ quota, IET asked them to proceed to the next subsection to ensure that they had the chance to address most of the target material. (The SRI SMEs had completed their work and performed reasonably well on this 25-TQ subsection.) Because of this gatekeeper KE intervention, the authors have by consensus excluded this subsection from results analyses below.

Our functional performance scoring is both manual and subjective. We employ multiple scorers with expertise both in knowledge representation and in biology. We have historically achieved highly consistent results by articulating specific, value-by-value scoring guidelines for all criteria against the following, relatively coarse, generic framework: 0—no serious effort evident/completely off-base; 1—mostly unsatisfactory; 2—mostly satisfactory; 3—(for practical purposes) perfectly adequate. To arrive at an overall score for functional performance on a given TQ, we: threshold scores for the last two ancillary criteria so that they do not exceed the highest score for (one of) the earlier, primary criteria; average scores for each criterion within a category; then average the category scores.

6 Experimental Results

6.1 Functional Performance Results

The major functional performance results are reflected in Table 1.

Team	*User type*	Representation	Answer	Explanation	Overall
Cycorp	*SME*	1.66	2.46	2.30	2.14
Cycorp	*KE*	2.54	2.58	2.56	2.56
SRI	*SME*	1.84	2.12	2.08	2.01
SRI	*KE*	2.09	2.48	2.40	2.32

Table 1: Means of teams' KEs'/SMEs' means of TQ scores

KEs' performance (the "gold standard" from RKF's perspective) was better than SMEs' with high statistical significance, but SMEs performed within 90% of the level achieved by their teams' KEs. We take the latter to reflect the relative effectiveness of teams' SME lab-fielded technology. There was no statistically significant difference across teams between the averaged scores of respective SMEs or KEs—either overall or at the criterion category level.

In a more detailed (unpublished/available upon request) treatment, we note statistically significant interactions among scores along the dimensions of individual SMEs, subsections, and question types in a categorization. All of these interactions washed out in the overall scores. We also note a "ceiling" effect, in that answer scoring with respect to several individual criteria exhibits large proportions of (highest-score) 3s. Elements likely contributing to this ceiling include our consistently accessible (i.e., low) quiz-level TQ difficulties and SMEs' consistent efforts to develop (supporting knowledge and) high-quality answers before moving on to additional TQs.

6.2 Economic / Reuse Results

Cohen et al. [2] profiled HPKB knowledge reuse as the fraction of knowledge items previously existing in a given context. We again have two main reuse contexts to explore: that of constants in axioms and that of axioms in the explanations/proofs of answers to TQs. (To economize, we include only the latter analysis.)

Axiom reuse results appear in Table 2 Results are given for each participant (designated by monikers). Each participants' (KB's) mean overall score and mean number of axiom occurrences used to answer a TQ are included here for reference. "UA" stands for "User-authored Axioms" and "PDA" for "Pre-Defined Axioms." "Used" indicates that the noted number of axioms actually appears in the explanation to one of the participant's answered TQs. "Unused" pertains to user-authored axioms that are not so used in a TQ (e.g., because the participant used them to author a subsection's target material before receiving its TQs). "Reused" pertains to user-authored axioms used to answer more than one TQ.

Table 2: Reuse data by SME/KE

Team	Type	Moniker	Mean Overall Score	Functional Performance TQ count	Reuse TQ count	UA used	UA reused	UA unused	PDA used	UA: reused / used	UA: used / (used+unused)	Used: PDA / (PDA+UA)
A	SME	Tweety	2.17	39	31	100	34	1881	103	27.16%	10.14%	47.65%
A	KE	MW	2.85	43	35	111	13	102	150	11.71%	52.11%	42.53%
B	SME	Amoeba	2.29	30	30	538	32	1355	304	15.81%	31.75%	62.41%
B	SME	Celula	1.68	25	25	143	58	490	465	73.71%	33.65%	27.99%
B	SME	Iflu	2.26	34	34	100	53	2211	323	41.89%	11.47%	36.20%
B	KE	PN	2.58	35	35	166	132	462	313	90.72%	54.94%	53.32%
B	KE	AS	2.50	34	34	296	106	1086	340	52.27%	33.50%	57.80%
B	SME	Vaccinia	2.59	35	35	254	163	1526	383	71.84%	17.75%	47.16%

Table 2 includes only one each KE and SME entry for Cycorp because of difficulties at evaluation time with KB instrumentation and later with information extraction. These reuse results are still incomplete, as may be noted by comparing the numbers of TQs answered/scored for Functional Performance and numbers of TQs scored for reuse. A further issue of note is that the Cycorp KE, cycMW, (legitimately) authored much general knowledge directly into Cyc, as pump priming, where it is counted as pre-defined rather than user-authored.

We present (in Table 2's last columns) three varieties of reuse percentages: of user-authored axioms that appear in more than one TQ; of user-authored axioms that appear (at all) in TQs; and of appearing pre-defined out of all appearing axioms. From an economic standpoint, we comment merely that the latter reuse rate seems (uniformly) sufficiently high to justify the claim that relevant prior content has significant benefit for KB development.

We had an additional motivation (beyond economics) to examine reuse of user-authored axioms across TQs. RKF's functional performance evaluation criteria, being TQ-based, could not address the generality of knowledge across different TQs. Evaluators were interested in quantitative metrics of cross-TQ axiom reuse as a hedge against unprincipled, one-shot axiom "hacks" without lasting value.

Table 3 reports numbers of TQ occurrences for each reused user-authored axiom.

Team	Type	Moniker	TQs > 32	TQs in [17 32]	TQs in [9 16]	TQs in [5 8]	TQs in [3 4]	TQs = 2
Cycorp	SME	Tweety	0	0	0	0	2	20
Cycorp	KE	cycMW	0	0	0	0	5	8
SRI	SME	Amoeba	0	0	2	2	5	86
SRI	SME	Celula	0	0	1	69	28	59
SRI	SME	Iflu	0	0	0	7	53	51
SRI	KE	sriPN	0	0	3	186	67	57
SRI	KE	sriAS	0	1	6	97	57	81
SRI	SME	Vaccinia	0	1	1	20	99	106

Table 3: Incidences of axiom occurrence counts across TQs

Superficially, high axiom TQ-incidences occurred much more frequently for users of SRI's SHAKEN tools than for Cycorp's KRAKEN tools. (The axiom TQ-incidence patterns for pre-defined axioms are qualitatively similar.) However, these data do not appear to indicate cross-team differences in knowledge generality. Cycorp SME Tweety's axiom TQ-incidence profile is quite similar to that of Cycorp KE cycMW whose work—with highly respected representations—received the highest mean overall functional performance score. Axiom TQ-incidence profiles are also similar across SRI's KEs and SMEs. We tentatively attribute the cross-team profile differences to: compactness of (arbitrary-arity) CycL relations compared to binary relations resulting from translating SHAKEN's frames for axiom-counting purposes (suggesting a scaling factor for axioms counted in a given TQ); and conceptual coarseness, compared to pre-defined predicates in Cyc, of SHAKEN's built-in relations (leading to greater applicability across TQs). Thus, we find no overall quantitative pattern indicating deficiency of appropriate knowledge generality for either team.

6.3 Subjective Qualitative Results

Deficiencies Identification: The KB Quality Review Panel concluded that SMEs, working alone, performed quite well at selected KB authoring tasks, but were less effective at others. SMEs with both teams were generally adept at placing and choosing concepts from the pre-existing ontology (i.e., they created and used knowledge at correct levels of specificity) and at general process description (i.e., they implemented Event-Actor vocabulary with accuracy and ease). The Panel highlighted as shortcomings in SME KBs the following major types: incompleteness, redundancy, and non-reusability. After describing these deficiency types below, we take up the question of their sources in the tools and in the KB authoring task.

Both teams' SMEs' KBs exhibit incompleteness, of three different kinds: content incompleteness (failure to describe a process fully, even where the textbook had); hierarchical incompleteness (failure to include natural siblings of a created concept); and interconnectedness incompleteness (failure to articulate obvious relationships between concepts).

SHAKEN SMEs' KBs exhibit significant redundancy attributable to limitations of the evaluated tools' inability to reason about authored concepts from the several distinct perspectives called for by different TQs. KRAKEN KBs also exhibit some redundancy. This usually is not of SME-authored knowledge, owing rather to re-creation by SMEs of pre-defined concepts.

Both teams' SMEs' KBs included concepts of suspect reusability. Mainly these were predicates, attributes, or concepts that combined concepts unnaturally—in a fashion that seemed difficult to reuse.

Suspected Deficiencies Sources: The Panel's and SMEs' combined attributions of the above-noted deficiencies to major tool and TKCP task sources including the following (in order of increasing challenge such sources seem likely to pose RKF tool providers): TQ- and textbook-focused SME orientation, absence or inaccessibilityof pre-defined knowledge, limited logical expressibility, and inherently difficult representation problems. We consider these in turn below.

Some KB incompleteness (especially of the content variety) are attributable to SME's attempts to tailor authoring in anticipation of unreleased TQs or in consideration of released TQs. (I.e., sometimes authoring favored TQ effectiveness over general applicability or reuse.)

That SMEs were explicitly directed to focus on authoring textbook content may explain some hierarchical and interconnectedness incompleteness.

Some of the above-noted deficiencies resulted from incompleteness in the pump-primed KBs that they received. SHAKEN did not allow SMEs to facet general collections into collections of different kinds of collection subtypes. A SME noted that this would have facilitated clearer hierarchical placement.

While KRAKEN SMEs had access to a substantial background KB and sophisticated representation language, this potential came at a price: access tended to be at times insufficient, during other times overwhelming, thereby limiting and even hampering SME productivity and expressive possibilities. Gatekeeper KE reports and SME surveys mentioned the labor-intensiveness of what turned out to serve as Cycorp SMEs' major axiom entry mode—browsing through existing axioms to discover one (with an appropriate predicate) to use as a template for editing and assertion.

Both teams' SMEs were—by design—somewhat limited in the logical forms they could use to express knowledge. SHAKEN SMEs were unable to make many assertions that deviated from the form $(\forall x (Ax \supset (\exists y) (B\ x\ y))$. KRAKEN users had access to more logical forms via a richer vocabulary of rule macro predicates, though interface issues again caused more general rule construction to be prohibitively difficult here.

A major indication frequently occurring in both team's KBs of inherently difficult representation problems is predicates lacking specificity, argument types, or supporting axiomatization. Another indication is impoverished versions of assertions whose formal representation would require complex logical expressions.

Feasibility Assessment: While it is clear that plausible near-term improvements to these tools (and their captured background knowledge) could address some of the above-noted shortcomings, it also seems (to the present authors) that KB authoring generally does include inherently difficult representation problems whose solution demands well developed logical skills and balancing different engineering principles. The ambition reflected in the present experiment to create tools that can empower a SME to full KB authoring independence—in arbitrary contexts—appears yet too grand.

While we have clear evidence that SMEs can author some high-quality knowledge in a sophisticated domain, we lack evidence that they can author high-quality predicates, analyze and refine background knowledge, develop rule paths to make sophisticated inferences work, or develop complex logical expressions required for some assertions. Also, it is not obvious how the existing tools could be refined to address such requirements.

Recommendation: We suggest that the KB development community's focus ought not be on tools that support KB authoring by "lone" SMEs (except where authoring tasks are relatively precisely defined and tools are fielded to support SMEs in a relatively mature authoring process). On the contrary, it should be on empowering SMEs to perform those KB authoring tasks they can be empowered to perform well. We believe the nascent RKF tools demonstrate a significant advance in such SME empowerment, and we recommend that in future experimental and developmental settings the relative strengths that SMEs and KEs bring to KB authoring should be exploited in a true "mixed-skills" team—a synergistic partnership.

We have some evidence that lightly trained SMEs are capable of significantly enhancing KE efforts to provide background knowledge that will be relevant to a KB authoring task. As a sequel to the TKCP evaluation, IET conducted a separate three-week evaluation intended to allow SMEs to explore teams' tools in a less structured setting. Eight (now tool-savvy) SMEs participated in an "expert knowledge" challenge problem (EKCP), pursuing KB authoring topics related to the life cycle of the Vaccinia virus—for which teams had authored no pump priming knowledge. An IET KE who had prepared some EKCP-supporting background knowledge development (in CycL) found that a Cycorp SME (who had not effectively authored Cyc predicates working alone) was readily able to contribute an informal specification that greatly facilitated the KE's work in extending the background knowledge to support the SME's needs.

We envision such interactions occurring throughout the KB authoring process, with SMEs and KEs contributing dynamically. The KE's role is always to perform sophisticated KB authoring tasks currently beyond SMEs' reach. We believe that the SME-feasible task set should expand naturally (in a "bootstrapping" fashion) over time, as the talents of SMEs are mined and new tools are developed to meet opportunities presented by existing tools and authoring processes.

7 Discussion / Conclusion

All styles of evaluation are useful in different contexts. Quantitative metrics are genuinely valuable for some purposes—e.g., inspiring a friendly competition among groups working in a common research initiative or demonstrating progress to an uninitiated, numbers-oriented supervisor. By far the long pole in the evaluation tent, however—from a system/process engineering, diagnostic point of view—remains subjective qualitative assessment. This is borne out by the comparative substance of our offered conclusions based on this activity and by the incorporation of insights and adoption of suggestions by technology providers working to develop the next generation of SME-empowering KB authoring tools.

We have seen that all three evaluation styles used here complement one another. The different quantitative metrics assist in each other's mutual interpretation (as, for example, when we appeal to Functional Performance in understanding Reuse), acting together as a synergistic set of reinforcements and consistency checks. We expect our effectiveness in the overall KB authoring enterprise to grow as the collective body of such techniques for understanding quality issues in KB artifacts, tools, and process continues to mature in a science of knowledge development.

8 References

[1] Alberts, B.; Bray, D.; Johnson, A.; Lewis, J.; Raff, M.; Roberts, K.; and Walter, P. *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell*, ch. 7, pp. 211–222, Garland Publishing, Inc., 1998.

[2] Cohen, P.; Chaudhri. V.; Pease, A. and Schrag, R. "Does Prior Knowledge Facilitate the Development of Knowledge-based Systems?" In *Proceedings of the Sixteenth National Conference on Artificial Intelligence,* pp. 221–2266, 1999.

[3] Cohen, P.; Schrag, R.; Jones, E.; Pease, A.; Lin, A.; Starr, B.; Gunning, D., and Burke, M. "The DARPA High-Performance Knowledge Bases Project." *AI Magazine* 19(4):25, 1998.

[4] Gruber, T. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," Technical Report, KSL-93-04, Department of Computer Science, Stanford University, 1993.

[5] Gruninger, M.; and Fox, M.S. "Methodology for the Design and Evaluation of Ontologies," in Proceedings of the IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, 1995.

9 Acknowledgement

This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) of the United States. This document has been approved for public release, distribution unlimited.

Appendix A: Cycorp's KRAKEN Tools

After over fifteen years of common sense knowledge base building, the Cyc project is well-equipped for Cyc to actively assist in its own extension. With its large set of common classes and instances, relationships and rules and knowledge contexts, Cyc has commenced its supervised learning process, pushing the envelope of what it knows. And, as learning occurs on the fringe of existing knowledge, leveraging and reusing this knowledge is key. This approach is realized in the KRAKEN system.[1]

The KRAKEN team metaphorically framed the task of extending a large knowledge base by viewing Cyc as akin to a child with limited proficiency in English. This (very young) "child" speaks CycL, a first-order predicate calculus-like language, as its "mother tongue" and has some knowledge of the common world. It has rudimentary notions of English, enough to verbalize most of its beliefs clearly, and can read simple English sentences with occasional help. In this view, the SME becomes a "teacher" who engages in a dialog with KRAKEN and exploits analogy, disambiguation dialog, and knowledge expectations to extend the system. Concurrently, the SME teaches the system how to express new information in English.

The KRAKEN team identified several key KB authoring tasks that KRAKEN could assist with: locating existing knowledge; adding knowledge through cut-and-paste techniques; fulfilling explicit knowledge expectations; reading simple sentences; deducing relations from examples; and assembling structured knowledge components (*e.g.,* non-trivial queries and rules) from short described scenarios. In addition, KRAKEN helps with correctness verification and strengthening of new knowledge.

For a KB of the size of Cyc, locating pieces of existing knowledge is a task in itself. At this writing, Cyc encompasses about 1.1 million assertions constructed from over 120,000 concepts and 5000 relations. Such dimensions make any "list-them-all" approach to searching impossible. However, Cyc also knows over 16,000 English verbs and nouns and over 2000 proper names and can therefore offer a natural language index into its knowledge. Once "within the vicinity" of particular concepts and relations, browsing is feasible. Additional organization is provided by Cyc's knowledge contexts ("microtheories").

Once a SME knows upon which pieces of knowledge to build, the KRAKEN system provides multiple ways for the learning process to proceed.

One principled approach is the explicit representation of knowledge expectation—*e.g.* knowing that when told of a new artist to ask for famous works by that artist. As new terms are introduced, KRAKEN will ask the SME concrete, salient questions. This approach is especially interesting, as the KRAKEN team is adding support for a SME to teach the system such knowledge expectations as well.

One major goal of the KRAKEN team has been to support KB authoring using simple English sentences. KRAKEN parses the sentences into an underspecified representation, which is then reformulated, based on the analysis of applicable argument constraints, into CycL. During reformulation, KRAKEN attempts to solidify the quantification, an aspect vital to knowledge engineering and highly ambiguous in natural languages. (Compare the class-level statement, "A dog is a mammal," to the instance level statement, "A dog is in the yard.") Like anyone learning English, KRAKEN asks for help when it gets stuck.

KRAKEN ensures—within bounds of reason—that the new information is semantically valid and neither in contradiction nor redundant with existing information. Even more important, KRAKEN attempts to fine-tune the strength of statements by suggesting ways to change their specificity or generality. Since stating knowledge at the correct level of generality requires mastering the available alternatives, KRAKEN guides the SME to subsumed or covering statements. This approach also exploits the human ability of recognition, instead of relying on recall.

No predicate set is ever complete, and KRAKEN provides the ability for the SME to define new relationships. The acquisition paradigm is structured around use cases: the SME provides KRAKEN with examples of how the predicate will be employed. This not only allows KRAKEN to compute the new relationship's argument constraints automatically but also jump-starts the population of the relationship and provides KRAKEN with believed suitable exemplars for communicating these relationships to other users.

For the assembly of more complex knowledge constructs, such as non-trivial queries and implications, the KRAKEN team has chosen an almost story-like approach: the SME lays out a scenario for KRAKEN, consisting of the involved terms and the relationships between these. Once the scenario has been "narrated" in this fashion, KRAKEN assembles the relationships and terms into a query or an implication.

---

[1] In order to achieve this goal, Cycorp teamed with Hans Chalupsky at the University of Southern Clifornia's Information Sciences Institute, Ken Forbus' Qualitative Research Group at Northwestern University, and the Artificial Intelligence Applications Institute at the University of Edinburgh to construct the KRAKEN system around Cyc.

Appendix B: SRI's SHAKEN Tools

The claim of the SHAKEN effort is that SMEs, unassisted by AI technologists, can assemble models of mechanisms and processes from components. These models are both declarative and executable, so questions about the mechanisms and processes can be answered by conventional inference methods (for example, theorem proving and taxonomic inference) and by various task-specific methods (for example, simulation, analogical reasoning, and problem-solving methods). A related claim is that relatively few components, perhaps a few thousand, are sufficient for SMEs to assemble models of virtually any mechanism or process. We claim that these components are independent of domain, and that assembly from components instantiated to a domain is a natural way for SMEs to create KB content.

The research in this project exploits and extends previous work in KBs, process description languages, qualitative physics, systems dynamics, and simulation. One scientific innovation is the idea of declarative and executable models (DEMs) assembled from components. The declarative aspect of DEMs supports conventional inference, whereas the executable aspect supports reasoning by simulation. For example, the declarative part of a model of aerosols is sufficient to answer questions like, "Will a 5-micron filter afford protection against this aerosol?" while the executable part is necessary to model the dispersal pattern of the aerosol.

The development of libraries of components made available to SMEs via restricted natural language based, graphical, or templatized interfaces is the principal means by which logic-oriented knowledge representation formalisms become accessible to ordinary users. Every modeling technology shows this progression: spreadsheets, finite-element packages, statistical packages, chemical synthesis software, Macsyma and Mathematica, architectural and CAD packages, graphics and HCI systems, *etc.* are accessible to ordinary users because they offer libraries of components. As a practical matter, then, it makes sense to provide SMEs with libraries of modeling components. As a scientific matter, we believe we can develop components that represent how humans think about mechanisms and processes.

The SHAKEN system has the following major functional components: a knowledge base, an interface for entering knowledge and asking questions, and a knowledge server.

The KB, also called the component library, contains a collection of components representing (1) general knowledge about common physical objects and events, states of existence, and core theories, including time, space, and causality, and (2) more specialized knowledge about microbiology and biological warfare agents. By a "component," we mean a coherent set of axioms that describe some abstract phenomenon (*e.g.,* the concept "invade") and that are packaged into a single representational unit.

The SHAKEN KB evaluated here contained roughly 250 components representing domain-independent events. These components would make copious use of core theories of time, space, and partonomy [7].

A graphical interface for knowledge entry enables a SME to assemble KB components. By "assembly," we mean the connection of components from the component library. The system evaluated here supports four basic operations: "connect," "specialize," "unify," and "add" [7]. Axioms are derived from the graphical representation, and the SME does not have to be trained in formal logic. The graphical representation is created by a combination of manual and automatic means.

The question-asking interface plays a central role in knowledge entry. A SME must be able to understand what is already encoded in the system, to locate components for assembly, and to ask arbitrary questions. SHAKEN returns answers in an easily understood format, and a SME is able to control the level of detail in an answer. SHAKEN as evaluated here supported parameterized questions—derived from a viewpoint grammar [6]—and similarity search. Presentation of answers to a SME is controlled using explanation design plans.

The knowledge server provides facilities for efficient storage and access, supports inference for answering questions and for assembly of components, and includes both general-purpose inference and special-purpose inference. For SHAKEN as evaluated here, reasoning support was provided by the Knowledge Machine (KM) representation system.

References for Appendix B

[6] Acker, L. and Porter, B. "Extracting Viewpoints from Knowledge Bases," in Proceedings of the Twelfth National Conference on Artificial Intelligence Annual Conference (AAAI-94), pp 547–552, 1994.

[7] Barker, K., Porter, B., and Clark, P. "A Library of Generic Components for Composing Knowledge Bases," in proceedings of the International Conference of Knowledge Capture, 2002.

A Task Domain for Combining and Evaluating Robotics and Cognitive Modeling Techniques

Nicholas L. Cassimatis, J. Gregory Trafton,
Alan C. Schultz, Magdalena D. Bugajska, and William Adams
Naval Research Laboratory, Code 5515 and 5513
4555 Overlook Ave. SW
Washington, DC 20375-5337

ABSTRACT

Building systems that integrate different artificial intelligence techniques to achieve a higher level of total intelligence is very difficult. In order to build integrated systems, simplifying assumptions or abstractions are usually made when working in a specific domain. As a result of these assumptions and abstractions, the proper evaluation of integrated artificial intelligence techniques can be quite challenging. We suggest that the domain of hide and seek is a particularly well-suited task for integrating robotics and higher-level reasoning mechanisms such as computational cognitive modeling. Three different instantiations of integrated systems in the "hide and seek" domain, which combine cognitive-level algorithms with lower level algorithms for perception and navigation are discussed.

KEYWORDS: *Robotics, Cognitive Architectures, Cognitive Modeling, Performance Metrics*

1. INTRODUCTION

How do we build intelligent systems, and evaluate the underlying algorithms? There are, generally, two main ways of building intelligent systems. The first possibility is to focus on a relatively small sub domain and build a system or set of algorithms that solve problems in that sub domain very well. Working in this manner generally leads to very efficient methods of solving a relatively narrow set of tasks. There are many researchers (including some of the current authors [1]) who build these kinds of highly specific systems. The second possibility is to build complex systems that solve a larger class of problems but that may be less efficient at each task and perhaps at the whole task. There are some researchers working on these kinds of problems (e.g. [10]), but building integrated systems is very difficult for a number of reasons.

First, the individual techniques are developed using different assumptions about their use (e.g., the input/output relationship with the environment they are in). Second, because most systems are designed for different domains, combining two techniques often involves modifying and adding new domain specific elements to the design. Finally, because each research group's assumptions and domain are so unique, each new group or project must reinvent machinery that is relatively incidental to their main interest. For example, a researcher trying to develop probabilistic reasoning techniques to aid in robot navigation must spend considerable effort acquiring and configuring a robotics platform with the appropriate sensors and actuators to test possible new techniques.

Each of the above integration issues makes it difficult to evaluate the effectiveness of any new technique or system. It is almost impossible to compare two techniques when one assumes video input from the environment and the other assumes sonar; or when one is designed for an office navigation task and the other for air traffic control.

Finally, the structural difficulties of integrating various techniques may encourage researchers to ignore or abstract away from difficult issues that hold back the field's progress. For example, there is concern [5] that when people working on "high-level" artificial intelligence techniques abstract away from perception and mobility issues, or when people working on perceptual and mobility techniques ignore high-level inference, they are actually ignoring the true substance of intelligence which lies at the interface between the two.

In Section 2, we present the task of hide and seek as a particularly well-suited task for integrating and evaluating artificial intelligence techniques. Section 3 describes three different instantiations of integrated systems, which combine cognitive-level algorithms with lower level algorithms for perception and navigation, and which use the hide and seek domain.

2. THE HIDE AND SEEK DOMAIN

In order to address these evaluation and integration issues, we are organizing a substantial amount of our research around the "hide-and-seek" task domain. This domain is forcing us to face the difficult integration problems between "high-level" cognitive architectures (for example, ACT-R [2] and Polyscheme [7]) and systems (such as SAMUEL [9]) for sensing and moving in a physical environment. In this section, we describe how using a robotic platform in this domain allows us to study a surprisingly wide range of issues in intelligence, perception and mobility.

2.1. Perception and Mobility

Agents that engage in hide-and-seek obviously cannot avoid the need to address a wide range of problems in perception and mobility. To find their targets, for example, agents must be able to identify (object recognition) and move towards their targets (path planning) without damaging the environment (obstacle avoidance). More generally, any information agents can gather perceptually will help them seek and navigate to their targets and the more efficiently they navigate to the target, the better they will be at hide-and-seek.

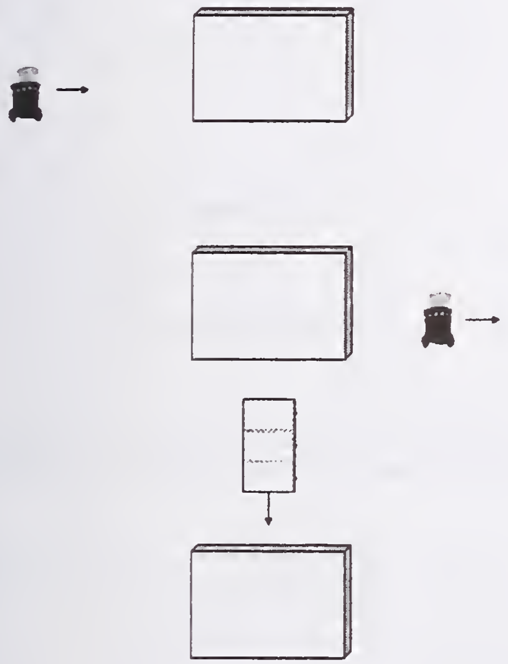


Figure 1. The robot that emerges from behind the screen can be the same robot that went behind the screen from the left because there was nothing behind the screen to block its motion. The barrier did not go behind the screen until *after* the ball did.

2.2. Temporal reasoning

In order to succeed at hide-and-seek, agents must perceive and reason about events that occur during various temporal intervals. The relations of those intervals among each other are important for predicting the outcomes of events and therefore the locations of objects that an agent might be seeking. Figure 1 shows a simple example of this. In the figure, a robot rolls behind an occluding screen and then a robot that looks the same rolls out. Next, someone places a large barrier behind the screen. Because the barrier was placed there *after* the robot-rolling event, you can assume that the space behind the screen was empty *during* the robot-rolling event and that the robot that emerged from the screen is the same as the robot that moved behind the screen.

Figure 2 presents the same scenario, except that the robot rolls behind the screen immediately *after* someone put the barrier behind the screen. In this case the robot that emerged from the screen cannot be the same as the robot the rolled behind the screen because this it did not have time to go around the barrier and it could not go through the barrier.

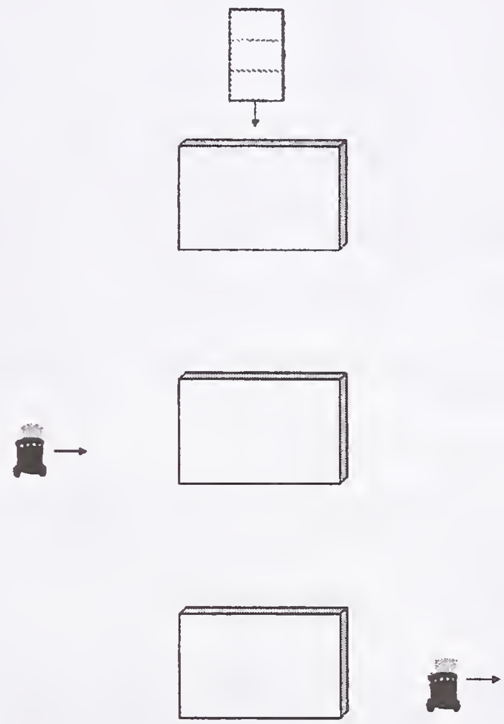


Figure 2. One knows that the robot that rolls out to the right is different from the robot that rolls in from the left because the barrier behind the screen would keep the left robot from rolling out.

In these very simple, illustrative cases and in more complex situations such as hide-and-seek, the task requires agents to make many temporal inferences in order to keep track of seeker or target agents and objects.

2.3. Logical deduction, falsification, default reasoning and explanation

Researchers using logical approaches to artificial intelligence have encountered many difficult issues regarding deduction, falsification, default reasoning and explanation, and they have constructed many sophisticated logical theories to deal with them. The following example shows that even the simplest physical interactions involve these issues.

In Figure 3a, a ball rolls towards a screen. In Figure 3b, it rolls behind the screen, but in 3c it fails to emerge from behind the screen and an object that blocked it is posited behind the screen. One can crudely formalize the inference that the ball should come out of the screen thus:

```
At(ball, left-of-screen, t1) ^
Moving(ball, right, t1) ^
Empty(behind-screen)
→
At(ball, behind-screen, t2) ^
Moving(ball, right, t2).

At(ball, behind-screen, t2) ^
Moving(ball, right, t2) ^
Empty(behind-screen)
→
At(ball, right-of-screen, t3) ^
Moving( ball, right, t3 ).
```

The inference that the ball emerges from the screen depends on the assumption that:

```
Empty(behind-screen).
```

When the ball fails to come out from the screen, you infer that the proposition,

```
Empty(behind-screen),
```

is not true and that there must be something behind the screen blocking the ball:

```
At(something, behind-screen, t2) ^
something != ball.
```

Many traditional issues from the formal logical study of intelligence arise here: what can you assume and why; what does it take to falsify an assumption; when there is more than one explanation for an event; which do you chose; etc. These are the usual issues surrounding explanation and default reasoning and they also occur whenever you try to build an effective hide-and-seek system.

2.4. Belief revision and reason maintenance

Any system that reasons in almost any nontrivial domain must often infer or assume facts that it must later revise. Because the system could have inferred more facts based on the originally assumed fact, revising its belief about the original fact is much more complicated than simply retracting it [8]. The system must retract all beliefs it inferred using the original fact that are not otherwise justified. Building systems that can revise their beliefs correctly has been a challenge for artificial intelligence researchers, even for those trying to build good models of common sense physical interactions.

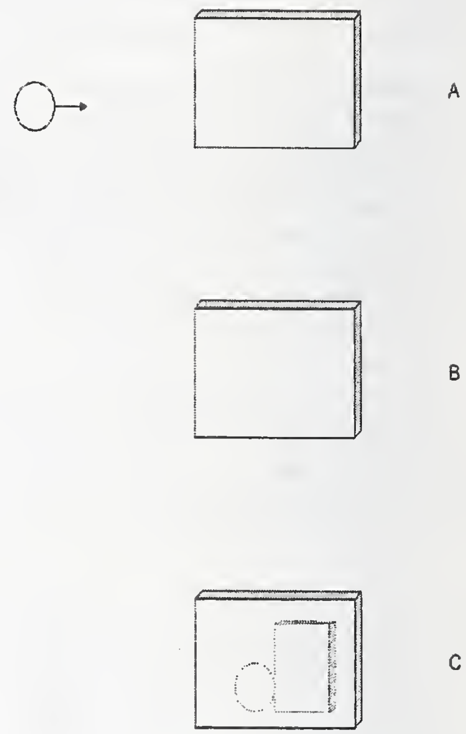


Figure 3. The ball rolls behind the screen (A), but does not roll out (B). There must be an object behind the screen that blocked it (C).

Consider an example. Figure 4a shows a scene where a screen occludes a table. A block is dropped above the table, it falls behind the screen and you infer that it comes to rest on the table. Then, when you are told that there is not just one table, but that there are two separated tables, as in 4b, you must revise your belief about where the ball went when it fell behind the screen. In this case, you assume it fell on the floor.

In general, in order to ascertain the location of any object, an intelligent system must make inferences about the object's location, which often depend on provisional information. For a system playing hide and seek, if the system spends time waiting for or attempting to acquire more definite information, the target would have more time to get away. Thus, any system that engages in the hide-and-seek task must be able to revise provisional beliefs and inferences that followed from it.

2.5. Planning, searching, problem solving

Events often have more than one possible outcome and systems can usually execute more than one action at any given time. The sequence of possible actions and/or inferences about event outcomes creates a huge "problem space" of possible world states and an intelligent system must choose a sequence of actions and/or inferences to achieve an adequate state.

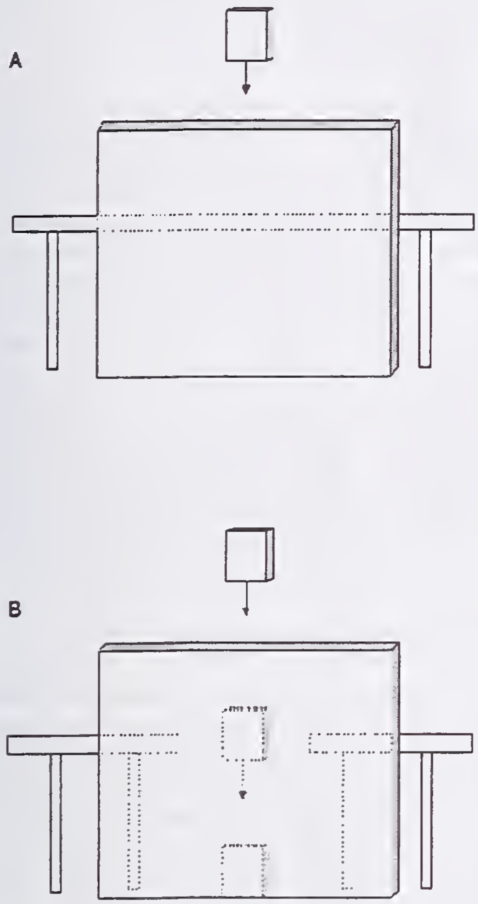


Figure 4. If, as appears in A, there is only one block behind the table, then you infer that it came to rest on the table. When you learn instead that there are two tables with a gap between them, as in B, you infer that the block fell through the gap and rests on the floor.

The need to search through problem spaces is most obvious in the hide-and-seek task when it involves robot mobility issues. Many algorithms for planning complex paths involve creating a visibility or region graph of the space and then searching the graph using traditional artificial intelligence search techniques.

Robots designed for the hide-and-seek task, however, need planning and search for much more than mobility alone. In the following example, we demonstrate that tracking the path of a simple ball can require searching through problem spaces that involve more than just the location of the ball. Figure 5 illustrates a simple physical interaction that requires backtracking search. Behind the screen in Figure 4 are two buckets. On the left, bucket A is filled with water and on the right, bucket B is full of hot coals. Figure 4 also shows a ball falling behind the screen. The ball is white and shaped roughly like a ping-pong ball, though it may be made of plastic or rubber. You see the ball fall behind the screen, though you neither see nor hear any further sights or sounds because the ball is too light to have dislodged anything and the screen masks soft noises. If your task is to figure out if the ball fell into bucket A or B, you might imagine that the bucket fell into bucket B and infer the consequences. To infer the consequence of landing in bucket B, you need to know if the ball is rubber or if it is plastic. You can imagine that it is rubber, infer that you would smell burning rubber, sense that you do not smell anything burning and therefore conclude that the ball is not rubber if it fell in B. Likewise, you can infer that the ball is not plastic if it fell into B because when you imagine a rubber ball lying in burning coals, you imagine a certain smell that you do not perceive. So if the ball fell behind B, it is neither rubber nor plastic. But you know it was one of these, so you know that the ball did not fall into B, but instead fell into A.

Similarly, the hide-and-seek task requires a broad array of search abilities.

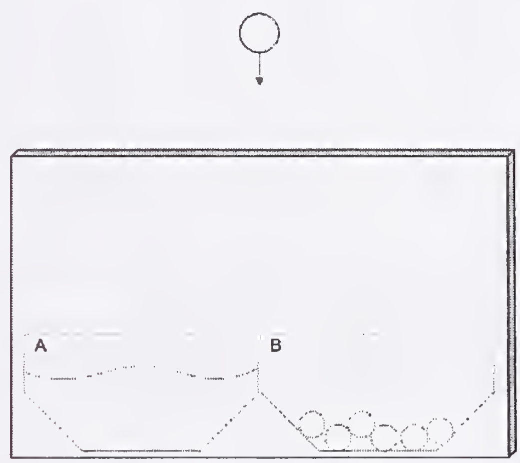


Figure 5. Bucket A is filled with water and bucket B is filled with hot coals. The ball falls into one of the two buckets.

2.6. Probabilistic inference

In many instances of hide-and-seek where events have more than one possible outcome, some are more likely than others. Seeking a target efficiently is often difficult when a scenario involves several possible series of outcomes, because the seeker must decide which of the many outcomes is most likely. Attempts to make such decisions are often called "uncertain reasoning" or "probabilistic inference".

Imagine an example like the last one, with the only difference being that you know more about the probabilities of each uncertainty. Bucket A takes twice the area of bucket B and the odds that the ball is plastic are 5:1. You are certain that you did not hear a splash, but are uncertain whether you smell any new smells. What are the odds that the ball is in A and what are the odds that it is in B?

This example shows that in order to keep track of the most likely positions of targets, agents must engage in probabilistic inferences. Similarly, hide and seek requires similar reasoning.

2.7. Social reasoning, communication and human-machine interaction

When a seeker agent is attempting to find an autonomous intelligent agent, it must be able to reason about that agent's mental state. Depending on the kind of target agent, this implies that competent behavior in the hide-and-seek task requires thought about emotions, beliefs, desires, personality traits, etc.

When a team of seeker agents is searching for a target, then each team member must be able to communicate with other team members in order to coordinate their behavior and execute a coherent search strategy. Depending on the communications abilities of the seeker agents, this could involve language use at all levels: speech recognition to decompose the acoustic signal from other agents into words, syntax and semantics to determine the meaning of the words and pragmatics to understand how the other agent(s) intend an utterance to fit into the larger joint seeking joint project.

When the seeking team contains both humans and robots, then the hide-and-seek task becomes a medium for studying robot and machine interaction with humans.

We believe that the full array of research issues in social reasoning, communications and human-machine interaction can be studied in the hide-and-seek domain.

2.8. Putting it all together

Our claim is that hide and seek is an excellent domain with which to study intelligence using integrated systems. We have presented several needed behaviors that seeker and target agents must have. We now go through a simple example of playing hide and seek to outline where each of these behaviors is needed. Bolded phrases below correspond to the behaviors we have discussed earlier.

Let us assume that we will play hide and seek with a robot. The robot will be the seeker agent ("It") first and search for the target in a room full of boxes, tables, and desks. The target initiates the game with the robot by talking to it (**communicating with it, using human robot interactions, probably language generation and language comprehension**). Next the robot counts to 10 and starts searching for the target agent. The robot must move around the environment in the room while avoiding obstacles (**perception and mobility**). The robot may also draw on its past experience playing hide and seek to determine that some places are better to hide behind and search those places first (**probabilistic inference**).

If the robot searches behind a box first, it may then assume that the target agent will not be there later (unless it moved) (**temporal reasoning**). If the robot then searches the entire room and does not find the target agent, it may reason that the target must have moved to a different (previously searched) hiding place while the robot was searching for it (**logical deduction, falsification, default reasoning, and explanation**). The robot must then decide to search the room again and re-check positions that it had already searched (**belief revision and reason maintenance**). The robot may decide that if the target agent can move around, it should search the room in such a way that allows it to see the maximum (or most likely) places the target agent would move to (**planning, searching, problem solving**). Once the robot finds the target agent, it must tell the target that it was found and perhaps even give it some feedback on how good the hiding behavior was (**social reasoning, communication, and HRI**).

2.9. Hide and seek in the real world

Many real world tasks are instantiations of the basic hide and seek domain. In the military, there are missions that directly require these skills, including ISR (Intelligence, Surveillance and Reconnaissance), RSTA (Reconnaissance, Surveillance, and Target Acquisition), and special operations including concealment. In non-military domains, hide and seek can be found in areas as diverse as Urban Search and Rescue, and inspection of facilities (e.g., searching nuclear containment facility and superfund sites).

3. INTEGRATED SYSTEMS

We now describe the three intelligent systems where high-level algorithms in the form of computational cognitive models are integrated with low-level perception and mobility algorithms, and which use various hide and seek domains. Available results will be briefly described (and the reader directed to appropriate papers for complete results).

3.1. A hybrid reactive/cognitive architecture for micro-air vehicles

We have developed a hybrid cognitive-reactive system that combines more traditional reactive, stimulus-response (S-R) behaviors with cognitive models [4][12]. In this work, we merge a cognitive model and a reactive system into a control system for autonomous vehicles. For this study, the system integrates SAMUEL, an evolutionary algorithm-based rule learning system [9] with ACT-R, a computational cognitive architecture [2]. In our hybrid system, the learning algorithm handles reactive aspects of the task and provides an adaptation mechanism, and the cognitive model handles the higher-level cognitive aspects such as planning and reasoning. The cognitive model also provides cognitive realism of the behavior.

Our hybrid controller was implemented for a simulated distributed micro air vehicle task. In the MAV task, group of vehicles cooperate to perform reconnaissance and surveillance, a version of the "seeking" task. We assumed each vehicle could detect certain ground features below the vehicle and obstacles, including other MAVs, within a defined range to the dies of the MAV. As a group, the MAVs needed to maximize the information gain about the ground features, concentrating on areas of more importance, and minimizing duplication of effort. In previous work, we successfully used genetic algorithms to evolve MAV control rule sets that could accomplish the above surveillance task [3][13].

The cognitive model implemented in ACT-R was based on the data collected during human-subject experiments performed at NRL and described in greater detail in [12]. In those experiments, the human operators would control the MAVs by directing them to goal locations using a point-and-click interface to the simulator. In this study, ACT-R, just like a human operator, was responsible for providing 2D goals to individual MAVs based on the current perception of the world. ACT-R's perception of the environment was closely matched to the perception of the human operator. ACT-R could "see" the position and state of all MAVs, and the position and value of discovered regions of interest.

SAMUEL was used to evolve stimulus-response rules to perform the collision-free reactive navigation behavior for the simulated MAVs. Each MAV used the same behavior evolved by SAMUEL in conjunction with the goals provided by ACT-R to safely navigate to a specified location. The current MAV sensor information is mapped to the conditions of the stimulus-response rules. The action of the rule that is activated specifies the action of the vehicle.

We found the performance of the hybrid controller to be comparable to the performance of the human controller, while allowing more vehicles to be controlled with fewer collisions. The model seems to capture some of the human's behavior and performance, while it also allows for higher levels of reactivity, which the humans were not able to handle. This suggests that our hybrid system is adequately modeling the humans' high-level cognitive functions, and also the difficult low-level reactive aspects.

3.2. Polyscheme

In order to study how to integrate multiple, seemingly incompatible, inferential and representational techniques, we used the Polyscheme cognitive architecture to develop the S6 system [7] that reasons about simple physical events that it perceives. This was especially helpful in understanding how high-level inference and planning techniques might combine with and help perceptual algorithms.

S6 views interactions in a simple physical world through a 2-dimensional projection of that world. S6 keeps track of the identity of objects, infers the character and existence of events it cannot see, predicts the outcome of events, explains events and nonevents and revises its inferences when it receives new information. S6 successfully reasons about many scenarios researchers present to infants and young children in order to study their knowledge of the physical world.

S6 combines specialized representation and inference techniques for identity, time, events, causality, space and paths to successfully deal with a wide range of situations. The knowledge representation schemes S6 uses include scripts, frames, logical propositions, neural networks and constraint graphs. The inference schemes S6 implements include script matching, rule matching, backtracking search, neural network propagation and counterfactual reasoning.

3.3. A learning cognitive model for playing the game of hide and seek

In our efforts to add cognitive models for higher level reasoning to traditional mobile robotics control, and to demonstrate the idea that more effective human-robot interactions are possible by using these computational cognitive models, we are modeling hide and seek behaviors in people, and using these to control a robot.

We have built a simple computational cognitive model of hide and seek. The model is based on a case study of a 3.5 year old learning to play hide and seek, specifically the learning that occurs as the child learns good and poor places to hide. The computational cognitive model is built within the ACT-R framework [2] and models the reasoning the child goes through as she plays and tries different hiding places. This is a very difficult task because there is very little feedback, very few trials, and very few suggestions.

The child went from hiding in a room with her eyes shut ("if I can't see you, you can't see me" strategy) to hiding under an upholstered chair in a different room. The model currently captures aspects of the child's learning by building a schematic representation of hiding and learning that some places are good to use as hiding places (e.g., under is good if the object is opaque; hiding under a piano is bad). The model also uses a simple ontology to reason about hiding given few suggestions ("Do not hide out in the open") and limited feedback ("You hid in a good place"). We are currently in the process of putting this computational cognitive model on a robot to play a simple game of hide and seek.

On the robot, perception is handled with a simplified perception model. Each object in the room is "labeled" with a color target whose color indicates the class of the object (e.g. desk or chair), and whose size indicates the approximate size of the object (so that the model can determine if it is big enough to hide behind. A color camera and a color blob detection algorithm [6] are used to find suitable objects to hide behind.

Low-level mobility of the robot is handled by a system that combines reactive navigation and collision avoidance, explicit path planning, map learning, and localization. This system is described in detail in [11].

4. GENERAL COMMENTS, FUTURE WORK

The proper evaluation of integrated artificial intelligence techniques can be quite challenging. In this paper, we presented the domain of hide and seek as a particularly well-suited task domain for evaluating the integration of low-level, reactive algorithms with higher-level reasoning mechanisms. Three different instantiations of integrated systems that combine cognitive-level algorithms with lower level algorithms for perception and navigation, were described.

We continue to push the integration of computational cognitive models into our systems. We believe that incorporating cognitively plausible behavior will permit more natural interactions between humans and robots. Using computational cognitive architectures and cognitive models can ease the ways in which robots communicate with their human team members, and vice versa. We have been exploring the addition of cognitive models for two goals. First, allowing the robot to use the same representations and qualitative reasoning as the human will allow for more effective and efficient communication. Second, endowing the robot with behaviors based on cognitive models of human performance allows the robot to exhibit behaviors that are similar to how a human might perform a task, thereby enhancing social human-robot interaction. Not only does this improve interactions with the robot's human team members, but is also critical for robots that need to interact with bystanders. We will test these arguments in future research.

5. REFERENCES

[1] Altmann, E. M. and J.G. Trafton (2002). Memory for goals: An activation-based model. Cognitive Science, 26, 39-83.

[2] Anderson, J. R. and C. Lebiere (1998). The Atomic Components of Thought. Mahwah, NJ: Lawrence Erlbaum.

[3] Bugajska, M. D, A. C. Schultz, J. G. Gregory, S. Gittens, and F. Mintz (2001). "Building Adaptive Computer Generated Forces: The Effect of Increasing Task Reactivity on Human and Machine Control Abilities." In Late-Breaking Papers Proceedings of GECCO-2001, San Francisco, CA.

[4] Bugajska, M. D., A. C. Schultz, J. G. Trafton, M. Taylor, and F. Mintz (2002). "A Hybrid Cognitive-Reactive Multi-Agent Controller." In Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), EPFL, Switzerland; September 30 – October 4, 2002.

[5] Brooks, R. "Intelligence without representation". Artificial Intelligence, 47:139--159, 1991.

[6] Bruce, J., T. Balch, and M. Veloso (2000). "Fast and Inexpensive Color Image Segmentation for Interactive Robots." In Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Japan; October 2000.

[7] Cassimatis, N. L. (2002), Polyscheme: A Cognitive Architecture for Integrating Multiple Representation and Inference Schemes. Ph.D. Dissertation, MIT Media Arts and Sciences Section.

[8] Doyle, J. 1979. A truth maintenance system. Artificial Intelligence, 12:231-272.

[9] Grefenstette, J., C. L. Ramsey and A. C. Schultz (1990). "Learning sequential decision rules using simulation models and competition," Machine Learning, 5(4), 355-381, October 1990, Kluwer.

[10] Lenser, S., J. Bruce, and M. Veloso. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup-2001: The Fifth RoboCup Competitions and Conferences*. Springer Verlag, Berlin, 2002.

[11] Schultz, A., W. Adams, and B. Yamauchi (1999). "Integrating Exploration, Localization, Navigation and Planning With a Common Representation," Autonomous Robots, v6 n3, May 1999; IEEE Press, pp. 293-308.

[12] Trafton, J. Gregory, Alan C. Schultz, Magdalena D. Bugajska, Shaun Gittens, and Farilee Mintz (2001). "An Investigation of How Humans and Machines Deal with Increases in Reactivity." Proceedings of Tenth Conference on Computer Generated Forces and Behavioral Representation (CGF&BR), Norfolk, VA; May 15-17, 2001.

[13] Wu, A. S. and A. Agah (1999). "Evolving Control for Distributed Micro Air Vehicles," proc. Of the IEEE Conference on Computational Intelligence in Robotics and Automation, IEEE, 174-179

Modelling, Self and Consciousness: Further Perspectives of AI Research

Ricardo Sanz
Universidad Politécnica de Madrid, Spain
Alexander Meystel
Drexel University, USA

Abstract:

Sound measurement of intelligence cannot be reduced to just measurement of performance. It is necessary to measure the real capabilities of the behavior generation engine of machines to be able to determine with precision their suitability for any particular task. We will see that it is necessary to focus on the architecture of the systems. The paper will present a summarial description of inner machinery of intelligence and how this architecture can serve as the basement for higher mental functionality. This will lead us to the formulation of a theory of conscious behavior and a the proposal of a research program focused into the nature and mechanisms of machine consciousness.

Keywords:

Intelligence, performance, mental architecture, self, consciousness.

1 Introduction: Measuring What ?

As Lord Kelvin said, "*to measure is to know*" and hence the importance of measuring intelligence to know better about it. Obviously, we know that the problem is not easy just from the very beginning: Is "Measuring of Intelligence" possible ?

In our search for machine replacements of humans in boring, dangerous or economically unviable activities, we use to check performance of machines [$Mind_{\mathrm{M}}$] against performance of humans [$Mind_{\mathrm{o}}$] or cognitive models of humans [$Mind_{\mathrm{d}}$]. However, do we sufficiently understand ourselves [$Mind_{\mathrm{o}}$]? or do we sufficiently understand the systems we design, manufacture, and use [$Mind_{\mathrm{M}}$]?

Generally speaking, we can consider two ways of measuring intelligence: with regard to a particular task or independently of any particular task (Pease [1] refers to this last form as *a priori* intelligence).

Intelligence manifests itself in the autonomous successful performance of tasks. Or, to be more precise, in the autonomous successful performance of a *task* by a specific *agent* in a concrete *context* [2].

```
Autonomy[AGENT,TASK,CONTEXT]
```

Extending the base idea of measuring *a priori* intelligence we need to measure this faculty independently of the concrete task, the concrete context and the concrete agent; otherwise what we will have is a concrete, particular measure, not very helpful to compare systems with a wide application domain (this being the case of conventional IQ tests, that just measure the capability of performing these tests and where extrapolation of results to other activities is highly risky).

To be able to obtain a measure of pure (*a priori*) intelligence we need to eliminate from the equation such factors as concrete bodies, concrete tasks and concrete contexts. This will leave the pure essence of intelligence. This vision of intelligence matches our intuitive, abstract notion of intelligence as a central faculty independent of particular factors that surround the activity. How can this possibly be achieved ?

In this paper we will try to identify the core essence of intelligence to be able to *directly* measure its capabilities instead of measuring the result of these capabilities in a concrete task. The conclusion of this identification will lead us to a research program that re-gains that old dream of artificial intelligence: building conscious machines.

2 Architecture and Performance

Presuming a functional equivalence of basic building materials, all our theories of mind [$Mind_{\mathrm{o}}$], [$Mind_{\mathrm{M}}$] and [$Mind_{\mathrm{d}}$] lead us to the concussion that only mental architecture can account for intelligent systems performance.

The perceived intelligence is strongly correlated with success. Intelligent systems architecture is a critical factor for success in the performance of any task [3]. Architecture is hence the point to focus our search for an a priori measure of intelligence [4]. Bad architectures lead to non performing systems.

Ignoring sterile differences between reactive and deliberative intelligence, all these theories do constitute interpretations that depend critically on representation of goals, states, contexts and bodies [5].

That representation is a central factor of intelligent performance has been know for decades. Execution engines do exploit representations to derive agent's actions. These execution engines receive varying names depending on the concrete task at hand: planners, behavior generators, predictors, etc. All them exploit the information about the world stored in a model (a world model) to derive actions. Minds are control systems based on models.

This leads to a typical architectural pattern for representation-based control system: the elementary loop of functioning *perceive-represent-plan-act* that is used as an elementary building block for more complex architectures (see Figure 1).

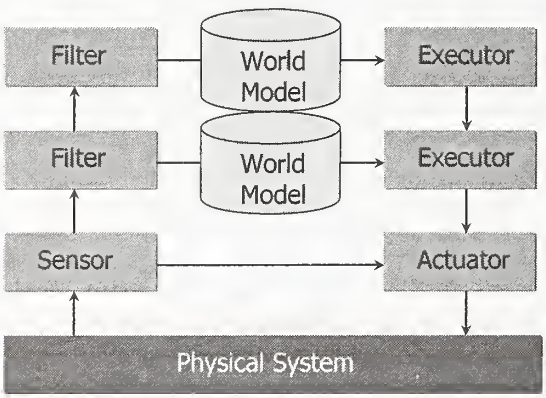


Figure 1: A basic, two layered, model-based control architecture. Each layer constitutes an elementary loop of functioning.

The effectivity of a concrete pair of components [*model,engine*] depends on the particular factors for autonomy mentioned before: *task*, *context* and *agent*. This means that a concrete pair, for example *[ordinary differential equations, Runge-Kutta simulator]* can be better than other pair *[first order logic predicates, resolution engine]* for a concrete task, for example *tank temperature prediction*, in a particular context, for example a *well-engineered refinery*, for a specific agent, for example a *model-based predictive controller*.

In many cases, this specificity lead us to sacrifice generality when dealing with constraints to attain specific execution properties (speed, robustness, cost). For example, mutiresolutional representation and control hierarchies offer cost effective solutions with bounded resources; for speed enhancement, compiled representations and engines adapted for them are employed.

Generality, however, is an extremely desirable property for a pair *[representation, engine]*. Generality is the mark of pure intelligence. Tradeoffs do obviously exist and have been used - mistakenly- as arguments against the suitability of general representations for the construction of intelligent agents [6].

Generality is out of questioning, however, because if we want to give to our systems control mechanisms with a high degree of *a priori* intelligence we need generality to overcome the barriers of the three factors: task, context, agent.

The broader the set of solvable tasks the greater the intelligence of the machine. This was that old dream of the Ultimate Problem Solver. For example: a washing machine is more intelligent if it is also able to minimize water consumption.

The greater the context-independence of the controller the higher the intelligence of the machine. This means that the controller can reach its objectives in a variety of execution contexts, *i.e.* is robust against variations in its execution environment, being able to handle uncertainty in a proper way. For example, a transelevator in an automated warehouse is more intelligent if it can avoid people eventually obstructing its way.

And last, machines that tolerate alterations in their own bodies and still fulfil their objective are more intelligent. For example, a car controller that can maintain car stability with a broken tire is more intelligent.

3 Systems with Self

Evolutionary pressure has forced a race-of-brains in the biosphere. One of the highest advances is when systems become able to extend its own capabilities. This means that the system can go beyond current engines and representations, current contexts and tasks, and even its current body. The highest levels of intelligence are those that not only do learning (enhancing engines) but also modelling (enhancing representations).

Intelligent natural systems do learn autonomously, or are provided externally the units of knowledge that are required for their successful functioning. This last approach is simpler in the case of artificial systems with limited scope. Anticipatory systems [7] can do all this process autonomously. The results of knowledge acquisition are models that capture reality (with the precision and level of resolution needed for the task). These models can be —easily— organized into automata models and hence the usefulness of computers to implement intelligent systems.

Truly intelligent systems do have models of their world and the tasks they perform and are able to enhance them. In fact, this is what all the business of science is about. Better models of reality to overcome, with our technologies, all the barriers

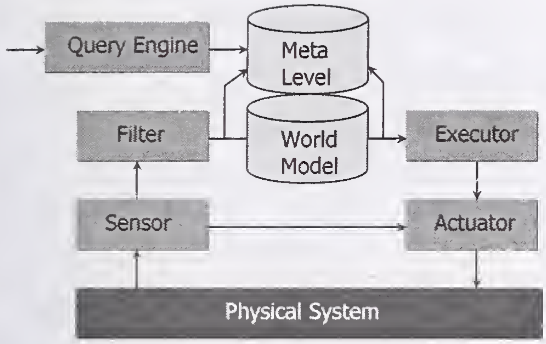


Figure 2: A not so basic, two layered, model-based control architecture. The metalevel provides introspection over an elementary loop of functioning.

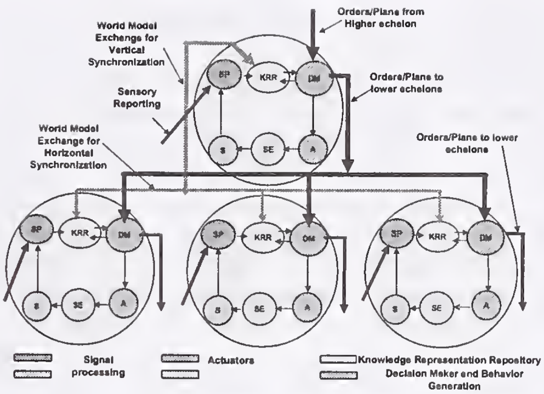


Figure 3: Another not basic, layered, model-based control architecture. The multiresolutional heterarchy goes beyond the single elementary loop of functioning into a collection of interacting loops.

from the past. Based on our better models of the external world (the context) we can do things (tasks) that go far away of what our grandfathers were able to do (with a similar body)[1].

The last step is easy to see: Truly intelligent systems do also model themselves. True intelligent systems maintain continuously updated, continuously enhancing representations of themselves. True intelligent systems are self-conscious.

Using this internal representations of themselves, intelligent machines can use their reasoning engines to reason about themselves and act accordingly. This representation and reasoning do also include more basic representation and reasoning processes (see Figure 2); intelligent systems do have metalevel representations and reasoning systems that, coupled with a query engine and a language interface, are used to interchange mental states with others: the states of the representations and reasoning processes (see Figure 3).

This simple analysis capture a commonsense thought: To be truly effective, intelligent systems need to be aware of themselves, i.e. need to be self-conscious. The nature of the self is this continuous perception and control of the body of the agent (see Figure 4). This is *the ghost in the machine*.

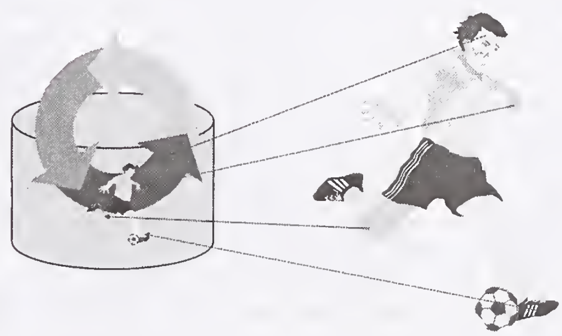


Figure 4: The nature of the self: the model of the body inside the model of the world and a loop of model-based control over it.

4 Autonomous Performance of Systems with Self

Automata models used by intelligent systems consist of explanations of the environment, states of the system and the appropriate rules of action. The basic process implemented in the loop of functioning is reproduced in Figure 5.

This model of reality need not be unitary but composed by a collection of elementary models. The set of all these models is aggregated together by the representation system of the agent.

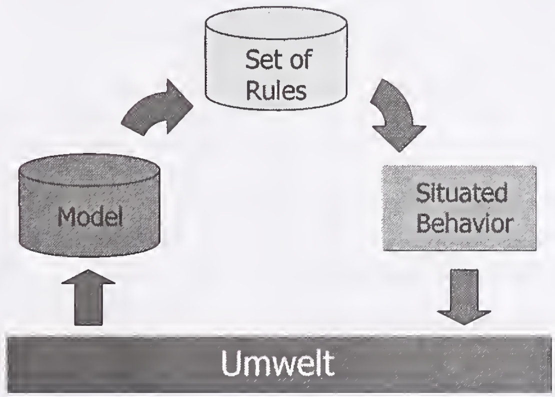


Figure 5: The process of modelling the world to behave inside it generates a semiotic loop.

[1]Consider, for example, the increase in life expectancy in the last fifty years based on better models of human inner workings.

Representation as a system of models appears in both natural and constructed systems.

This representation system also includes intelligent system goals that are distributed across the heterarchy of models of reality. In a similar way, we can think that autonomy is distributed over the architecture with the same level of granularity. Each level has a component of autonomous behavior determined by one or many goals of this level and the corresponding performance measures, and as the resolution becomes lower, both behaviors and goals are generalized. Autonomy is embodied in the hierarchy of goals and performance measures supported by internal models. In some sense, we can say that automata models have a rudimentary form of free will.

The intelligent system has a certain degree of autonomy. However, its activity is oriented toward the goal of the larger system (lower resolution unity that the intelligent system belongs to).

Consider a bee. While everybody would agree about its autonomy, no one will doubt that it is —the autonomous activity— oriented toward the goal of the swarm. At least, it must not contradict it in the long term. We also can talk about the autonomy of a can-picking robot. But one should agree that its autonomy is oriented toward the goal implanted into this robot by the designer —a sort of lower resolution level.

In the case of artificial systems, this dependency between levels of autonomy is so strict, that we strive only for *bounded autonomy* in the design of these systems.

Actually, the mechanisms of autonomy in the overall system as well as functioning of autonomic subsystems determine the viability of the system.

At each level of resolution, one can talk only about the degree of autonomy enclosed within the goal-oriented activity. This degree is determined by the self of this particular level. The degree of autonomy is required to cope with the high-resolution eventualities that at the lower resolution level are parts of the uncertainty taken into account.

The question *Why do we need systems with "self" ?* has a clear answer now: Having a self (a continuously updated model of agent's body) do increase intelligent systems performance.

To be more concrete, this increased performance can be shown by the collection of tasks that depend on this schema of representation and control:

- Introspection/reportability
  - Optimi zation (reorganizing own inner processes)
  - So cialization (collaborating with other agents)
- Fault handling
  - Fault detection and isolation (finding problems)
  - Diagnosis (identifying causes)
  - F ault management (devising walkarounds)
- Autonomous behavior
  - Re goaling (changing tasks)
  - Reconfiguration (changing body)
  - T ooling (changing context)

5 Intelligence, Self, Consciousness

In this process of intelligent systems analysis there are many objectives. In some sense we are stepping toward $[Mind_o]$ by means of implementations of $[Mind_M]$ that provide progressively accurate behavior. To our understanding, the architectural model presented here is a good unified theory of natural and artificial intelligence systems.

This theory do explain some of the more difficult observed aspects of human minds, while avoiding entering into the non-implementable field of metaphysics.

One of the more puzzling aspects for human mind is the uniqueness of "self". As we have seen, the model of the extended-plant (the body+the controller) can be single or multiple, not necessarily unitary. The question is: Is there any reason for the uniqueness of the "self"?.

Our hypothesis is that unitary selves provide evolutionary advantages (better autonomous performance). Having a single self enables resolution of autonomous control problems with scarce resources in the presence of higher degrees of uncertainty. The presence of the single self guarantees the coherence of the set of multiresolutional goals of the intelligent system.

In relation with $[Mind_o]$, there are no widely accepted explanations of conscious/unconscious behavior. Some authors distinguish between unitary and dual explanations (having one or two mechanisms for the conscious and the unconscious) and multiple theories are available on both sides (See [8] for a good survey).

One of the major problems is that while unitary explanations are more aestetically pleasant, they fail to provide the necessary qualitative distinction between the conscious and the unconscious that many authors want to see. The origin of the problem can be traced to the perceived distance between conscious and unconscious that from our point of view does not exist. Consciousness is not a binary property, it only looks like that because the interaction with external agents (with others) is performed only an a concrete high level of the control hierarchy.

The main obstacles for a unified theory of mind ($[Mind_o]$+$[Mind_M]$ are the chauvinism of human species and the manicheism of most theories, that appears everywhere: Representation/representationless, deliberative/reactive conscious/unconscious, symbolic/subsymbolic, biologic/beer can.

There are no intelligent systems without representation; every system that has a sensor has representation. Reactive control systems are just degenerate cases of deliberative control systems (where deliberation is reduced to a simple I/O mapping).

There is even disagreement about what is *consciousness*. Some authors distinguish three types: Access consciousness, reflective consciousness and phenomenal consciousness. There are authors that distinguish upon seven types !!!.

The key for the emergence of self-consciousness is integration of information about the body with information about the world (see Figure 6)[9].

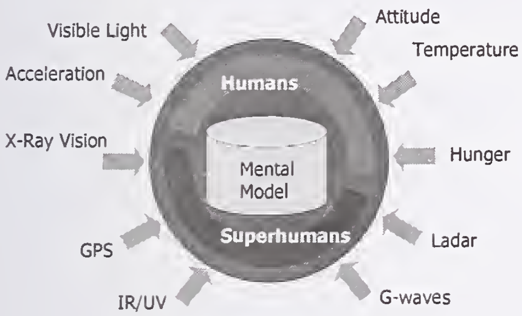


Figure 6: The process of integrating information from incoming sensors lead to progressive world-awareness. Systems with better sensors can be more aware if properly designed.

Consciousness increases as more information (from the outside, from the inside, from the mental processes, etc.) is integrated in a dynamical mental model that includes the self.

The -wicked- problem is not achieving consciousness but achieving human-like consciousness (i.e. being recognized as humans by other human minds) and this can be done only by means of human-like reportability and a human-like mental architecture. But building humans is nonsense from a practical perspective and recognizing consciousness in very alien systems is something that not everybody is prepared to do. As Thomas Nagel would ask: *What is it like to be a Tomahawk missile ?* [10]

6 Conclusions

Semiotic principles of computation provide a sufficient background for developing computer systems that demonstrate elements of self, consciousness, and free will.

The prerequisite for achieving this milestone in the intelligent computer systems development is focusing on interpretability of representations as the major factor of performance. This can be approached by developing multiresolutional (multigranular, multiscale) systems of knowledge representation equipped with a sufficient set of procedures to exploit the representations.

Eventually, the integration of these hierarchical model-based control loops will lead to ascertain the emergence of SELF.

We can conclude that *consciousness* is just an operational mode of a -not necessarily- complex controller. Being conscious is just having a running controller (being ON). Self-consciousness appears by the very same method when the sensed plant is the intelligent system proper.

As the multigranular system of semiotic closures emerges, it arrives at the phenomenon of representing and monitoring itself: self-consciousness. SELF [consciousness] is the multigranular system of semiotic closures constructed in representation for interpreting intentionalities of a system within its blended global multiscale coordinates. SELF is also supported by a multigranular system of goals that can be considered a provisional reminder of the results earlier produced by its intentionality system.

References

[1] A. Pease, "Evaluation of intelligent systems: The high performance knowledge bases and ieee standard upper ontology projects," in *Proceedings of the Workshop Measuring the Performance and Intelligence of Systems, PerMIS'2001*, (Mexico D.F.), September 4 2001.

[2] R. Sanz, F. Matía, and S. Galán, "Fridges, elephants and the meaning of autonomy and intelligence," in *IEEE International Symposium on Intelligent Control, ISIC'2000*, (Patras, Greece), 2000.

[3] J. Albus and A. Meystel, *Engineering of Mind: An Introduction to the Science of Intelligent Systems*. Wiley Series on Intelligent Systems, New York: Wiley, 2001.

[4] H. H. Hexmoor, "Smarts are in the architecture!," in *AAAI Spring Symposium*, 1995.

[5] A. Meystel and J. Albus, *Intelligent Systems: Architecture, Design, Control*. Wiley Series on Inteligent Systems, New York: Wiley, 2001.

[6] R. A. Brooks, "Elephants don't play chess," *Robotics and Autonomous Systems*, vol. 6, pp. 3–15, 1990.

[7] R. Rosen, *Anticipatory Systems*. Pergamon Press, 1985.

[8] J. G. Taylor, *The Race for Consciousness*. Cambridge, MA: MIT Press, 1999.

[9] R. Sanz, "An integrated control model of consciousness," in *Proceedings of the conference Toward a Science of Consciousness*, (Tucson (AZ), USA), April 8-12 2002.

[10] T. Nagel, "What is it like to be a bat?," *The Philosophical Review*, October 1974.

[11] M. Asada, E. Uchibe, and K. Hosoda, "Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development," *Artificial Intelligence*, vol. 110, no. 2, pp. 275–292, 1999.

[12] A. Sloman, "Virtual machines and consciousness," technical report, University of Birmingham, 2002. http://www.cs.bham.ac.uk/research/cogaff/.

[13] C. Joslyn, "Levels of control and closure in complex semiotic systems," in *Workshop on Closure: Emergent Organizations and their Dynamics*, (University of Ghent, Belgium), May 3-5 1999.

[14] N. R. Jennings, "On agent-based software engineering," *Artificial Intelligence*, vol. 117, pp. 277–296, 2000.

[15] N. Shadbolt, "The shape of things to come," *IEEE Intelligent Systems*, pp. 2–3, September/October 2001.

[16] A. Meystel and R. Sanz, "Self-identity of complex control systems," in *Proceedings of the Workshop on Theoretical Fundamentals of Intelligent Systems*, (Durham (NC), USA), March 11 2002.

[17] W. Dress, "A Bayesian approach to extracting meaning from system behavior," in *Proceedings of IEEE Systems, Man and Cybernetics Conference*, 1998.

[18] P. Horn, "Autonomic computing: Ibm perspective on the state of information technology." IBM Research, 2001.

PART I
RESEARCH PAPERS

General Discussion Panel 2

Information Access Technology: Does it Concern Mechanisms of Mind and Intelligence?

Information Access Technology: Does it Require Getting Involved in Mechanisms of Mind and Intelligence?

Martin Herman
Information Access Division
NIST

The world produces between 1 and 2 exabytes (10^{18} bytes) of information each year -- about 250 megabytes for every man, women, and child on earth. [Lyman, Peter and Hal R. Varian, "How Much Information," 2000, http://www.sims.berkeley.edu/how-much-info] Therefore better tools and technologies for information access and information management are needed to take full advantage of the ever-increasing amounts of digital information.

The discussion here will focus on technologies for accessing unstructured, digital multimedia and other complex information, including text, web pages, images, video, voice, audio, and graphics (both 2-D and 3-D). Examples of such technologies include search and retrieval techniques; information filtering techniques; methods for transforming speech, text, images, and video to representations that can be searched and filtered; user interaction techniques, including multi-modal approaches, that provide access to information; visualization methods that provide access to information; and sensor data acquisition and management. Note that these areas overlap what are considered to be traditional areas of AI, including speech processing and understanding, image and video processing and understanding, natural language processing and understanding, search techniques, data mining, text mining, speech mining, and video/image mining.

Generally, I agree with the notion that information access technologies can be improved through the use of AI. Although many AI technologies are currently not mature enough to be applied effectively, a program that advances these technologies could lead to better information access technologies. The following paragraphs discuss this further.

Traditional statistics-based information retrieval approaches seem to have reached a performance ceiling. New approaches that combine the linguistic analysis used in natural language processing with the statistics-based approach are showing great promise in improving performance. The AI approach could become particularly important for areas such as question-answering systems and summarization systems. In fact, from a functional point of view, any system that can look up information in databases and on the web and answer questions and generate summaries in a manner similar to humans is, by definition, an intelligent system.

The transformation of speech, text, images, and video to representations that can be searched and filtered requires a certain degree of "intelligence." In particular, speech-to-text and image/video understanding have traditionally been considered AI tasks. The extraction of sophisticated metadata from video and images is a particularly difficult task, and many would agree that extracting information such as objects, relationships among them, events, people, and their goals from video is a task that requires intelligence.

There are many different kinds of user interaction techniques. The traditional graphical user interface (GUI) used in much of today's software is not considered "intelligent," but neither are they easy to use by "non-technical" people. One approach to the promise of intuitive user interfaces is through perceptual user interfaces, those in which the machine perceives and understands what the user is doing, and the user can interact with the machine as if she is interacting with another human, through speech, natural language, and gestures. This would require the machine being able to perform tasks such as identifying and perceiving users and their actions and goals, and understanding and anticipating user needs.

In summary, I believe that the field of AI is still quite immature. However, the potential rewards to fields such as information access can be enormous if AI technologies can be significantly advanced. An evaluation-driven, metrics-based program in AI could significantly contribute to such a program.

Information Access: Do You Mind?

John Cugini, ITL

What's the Problem?

Successful execution of many information-based tasks depends crucially on contextual knowledge. Language processing is particularly sensitive to context and I will concentrate on it in this abstract as the example *par excellence* of knowledge-dependent information access.

The only type of system currently capable of deep semantic knowledge is the human mind. This system has two notable and perhaps related aspects: 1) knowledge does not appear to be *symbolically* represented in any straightforward way - no one expects to find logical propositions directly encoded in neurons. 2) human-based knowledge appears to be intimately connected to consciousness - a phenomenon about whose nature there is simply no consensus (e.g. see Daniel Dennett vs. David Chalmers). Several development paths (not mutually exclusive) for intelligent access suggest themselves:

Muddle Through

Does successful information access really require a rich world model to provide context, or can we get by with brute force (think Deep Blue) and/or statistical algorithms? Or perhaps we can get by with many "little" scope-limited applications (e.g. airplane reservations - in which the "world model" is essentially a database system)?

It seems doubtful to me that a truly robust speech recognition system (one with performance comparable to a human transcriber) or even a text-based question-answering system could be crafted in the absence of a more general knowledge base. It's one thing to play chess using special chips and brute-force; quite another to interpret English.

Formal Representation

If extensive knowledge is required, can it be represented in a symbolic, formal (and hence manipulable) way, a la CYC (see www.opencyc.org)? Is there a "critical mass" of such common sense knowledge? Must this system be updated manually or might it achieve NLP capability sufficient to let it learn new information (including the formation of new concepts) directly from text? Just because humans apparently don't represent knowledge this way doesn't mean that it can't be done (airplanes don't need ot flap their wings to fly).

It's difficult to predict how fruitful this approach will be. It at least takes the problem of common sense knowledge seriously. But it's a big world with a lot of facts - suppose you ask a friend about her round of golf and she e-mails back: "Oh, it was pouring for a while - the clubs were practically flying out of my hands." What would it take to build a system capable of concluding that her score was probably higher than usual? Or explaining why the clubs were "flying"?

Non-symbolic approaches

Symbolic systems also have a problem of anchoring: even if the knowledge base correctly encodes the ***relationships*** among concepts, there must be a set of base concepts that are not further defined linguistically. Although the symbolic approach enables reasoning at the propositional level, at best it can encode conceptual knowledge only by mapping words to other words.

But humans have, for instance, a irreducibly non-linguistic understanding of what "yellow" means; human conceptual knowledge is intimately tied to sensory and emotional states (consciousness rears its ugly head), which we can label ("yellow", "pain", "sorrow") but not further explicate. And surely there are many (most?) questions the answering of which requires such non-verbal understanding.

What are the prospects for building a system that associates sensory inputs with concepts? Computer-based "neural" networks can be trained to recognize textual characters - is this simply a good trick or does it scale up to recognition of a wide variety of objects?

Conclusion

Enough smart people have been trying for a long enough time that one may safely conclude that the current limitations of intelligent access are not accidental; it will take more than yet another clever algorithm to transcend them. A significant world modeling capability is necessary to achieve major improvements over the current state of the art. What the payoff curve looks like (system performance as a function of richness of representation) is one of the major open questions.

Using Speech Technologies for Information Access: Does it Require Getting Involved in Mechanisms of Mind and Intelligence?

John Garofolo
Information Access Division
NIST

Speech is arguably man's oldest and most natural form of communication. Speech and language are also inextricably linked to human thought and intelligence. Therefore, the recognition and understanding of spoken and written language was from the beginning an important component of artificial intelligence research. Initial efforts at speech recognition using classic AI techniques were thought to have failed because of the computational limitations of the time. Yet, even with the major advances that have occurred in computing power over the last decade, successful communication with machines using human language remains elusive.

While it's true that speech-based information systems (primarily telephone call centers) are being deployed on almost a daily basis, these systems are typically highly constrained to a particular limited vocabulary and/or discourse domain. If one tries to ask one of these systems a question that is outside of its narrow scope or if one uses a word that is outside of its vocabulary, one is quickly switched over to a human operator, or worse, sent into telephone oblivion. It's somewhat ironic that the current wealth of speech recognition software/services as well as their limitations are largely due to a mass decision by the research community to abandon traditional AI approaches in the mid-to-late 70s and instead focus on probabilistic methods.

When traditional AI approaches became intractable when applied to language, statistical approaches using variations of Hidden Markov models and neural nets were employed. For a time, during the late 80s and early 90s, there were huge improvements in accuracy. To work effectively, these approaches require large amounts of exemplary training data. So, large amounts of money were devoted in the research community to developing transcribed recordings of hundreds of hours of speech. While larger amounts of training data improved accuracy for particular domains, significant generalizable improvements in the technology were really not occurring. For a time, as progress began to stall, larger and larger training corpora – eventually in the hundreds of hours -- were employed. However, it was soon learned that the problem could not be solved with the amount of data that could be reasonably collected and transcribed and it has been suggested that many thousands of hours of speech would be necessary for significantly improved recognition using these techniques. Unfortunately, some researchers now believe that statistical approaches are inherently limited and that further progress down that path is unlikely. There are clearly at least three important components of recognition of speech by humans that are not addressed by existing approaches: robust acoustic modeling, explicit linguistic knowledge, and world/contextual knowledge. Because of these limitations, recognizers were (and continue to be) limited to the types of vocabulary they could recognize and the acoustic conditions within which they could work with any accuracy. Unfortunately, to date, the speech research community by itself has been unable to address these issues.

Yet progress could not have been gauged at all if the community had not had a way to evaluate recognition performance. Therefore, an important factor that cannot be overlooked in the development of speech recognition technologies was the development of evaluation methodologies and practices. In 1987, with DoD sponsorship, NIST began a series of speech recognition evaluations that continue today. Through the years, the evaluation domains have changed and become more realistic and challenging, but the approach and metrics have remained relatively stable. These evaluations use "canned" recordings of speech and are, therefore, repeatable. In the early years of speech technologies, only demo-based anecdotal so-called evaluations were performed and great claims were made which could not be substantiated. This brought about the "dark days" of speech recognition in the 70s when it was realized that speech recognition (as well as other AI technologies) had been "oversold". The NIST evaluations helped to reverse perceptions about speech recognition and demonstrated measurable progress over time as well as provided direct comparisons between systems. As such, these evaluations have also helped to propel the research into productive directions.

Recently, with NIST's assistance, the Defense Advanced Research Projects Agency (DARPA) has begun a new program to explore new approaches in speech recognition technology. The program has set out to improve performance in two ways: 1) By exploring novel approaches for word recognition; and 2) By creating integrated technologies generating enriched transcripts. This new area, deemed "Rich Transcription", is an effort at generating recognition output that contains a variety of metadata in addition to the words that were spoken. The resulting rich transcripts with speakers indicated, sentences marked, etc. can be rendered into human-readable form and are also more useful for other downstream processing technologies such as retrieval, information extraction, summarization, and translation. Further, the metadata can be "fed back" into the recognizer to improve the word recognition itself. This new program provides a "back door" back into the world of AI by integrating linguistic knowledge into the recognition process and by rethinking that acoustic and language modeling process. Since the recognition process will be tightly integrated with deeper linguistic processing to provide an understanding of the structure and meaning of the words that were spoken, this is an initial step back to speech recognition's AI roots. Other efforts at building speech understanding and discourse capabilities are taking a similar approach.

NIST is widening the scope of its speech recognition evaluations to measure the accuracy of the production of the Rich Transcription metadata as well as the words produced by the recognition systems. It is hoped that this new, broader approach will help to propel the recognition and understanding of human speech by machines to a new level which will fulfill one of the promises of AI.

PART I
RESEARCH PAPERS

Banquet Speech

In Quest of Performance Metrics for Intelligent Systems—A Challenge that Cannot be Met with Existing Methods

Lotfi A. Zadeh*

As we move further into the realm of intelligent systems, the problem of devising performance metrics for assessing machine intelligence looms larger and larger in importance. The problem is there, but does it have a solution? A somewhat unorthodox view which is articulated in the following is that (a) complete solution is beyond the reach of existing methods; and (b) that a prerequisite to solving the problem is a better understanding of a broader problem, namely, the basic problem of concept definability. To this end, what is presented in the following is a sketch of what may be called a theory of hierarchical definability, or THD for short.

In science, and especially in natural sciences and mathematics, there is a long-standing tradition of expecting that concepts be defined clearly and precisely. But as we move from the natural sciences to the sciences of the artificial, two basic problems came into view.

The first problem relates to the need to formulate our definitions in ways that can be understood by a machine. For example, if I command a household robot to take the dishes off the table, I must define what I mean by "take the dishes off the table." Or, if I instruct a machine to summarize a document, I must define what I mean by a summary. And, how can I assess the MIQ (Machine IQ) of a machine that executes my commands?

The second problem is that we encounter, much more frequently than in the past, concepts which do not lend themselves to precise definition. Among familiar examples of such concepts are intelligence, creativity, autonomy, adaptivity, relevance, robustness and causality.

We have been largely unsuccessful in formulating operational definitions of concepts of this nature. Why?

A view that is advanced in the following is that the primary reason for the lack of success is that the concepts in question, and many like them, are intrinsically fuzzy, that is, are a matter of degree. Thus, when we try to define such concepts within the conceptual framework of classical, bivalent logic, we encounter a fundamental incompatibility—an incompatibility between crispness of definitions and fuzziness of the concepts we try to define.

---

* Professor in the Graduate School and Director, Berkeley Initiative in Soft Computing (BISC), Computer Science Division and the Electronics Research Laboratory, Department of EECS, University of California, Berkeley, CA 94720-1776; Telephone: 510-642-4959; Fax: 510-642-1712; E-mail: zadeh@cs.berkeley.edu. Research supported in part by ONR Contract N00014-00-1-0621, ONR Contract N00014-99-C-0298, NASA Contract NCC2-1006, NASA Grant NAC2-117, ONR Grant N00014-96-1-0556, ONR Grant FDN0014991035, ARO Grant DAAH 04-961-0341 and the BISC Program of UC Berkeley

Viewed in a slightly different perspective, the problem relates to the inadequate expressive power of the definition languages which are at our disposal, namely, the natural language and the language of bivalent logic. What this implies is that, to solve the problem, we have to add languages with higher expressive power to our repertoire of definition languages. This is the basic idea that underlies the theory of hierarchical definability, THD.

In THD, the languages that we add are based on fuzzy logic since they must be capable of serving as definition languages for fuzzy concepts. More specifically, the definition languages in THD form a hierarchy represented as (NL, C, F, F.G, PNL), where NL is the lowest member in terms of expressive power, and PNL (Precisiated Natural Language) is the highest. It is understood that every member of the hierarchy subsumes those below it.

The C definition language is the language of mathematical analysis, probability theory and bivalent logic. This is the language that we learn when we take courses in mathematics, probability theory and logic. The F language is the language of fuzzy logic without granulation, and the F.G language is the language of fuzzy logic with granulation. PNL (Precisiated Natural Language) is fuzzy-logic-based language with maximal expressive power.

A simple analogy may be of help. In my progression of learning, I start with my knowledge of a natural language. After entering a university and taking courses in mathematics, I add to NL my knowledge of C. At this stage, I can use the union of NL and C as a definition language. Then, I take a course in fuzzy logic. In this course, first I learn F, then F.G and finally PNL. At the end, I can use PNL as a definition language, with the understanding that PNL subsumes all languages below it in the hierarchy.

What is PNL? The basic idea in PNL is that a proposition, p, in a natural language, NL, may be precisiated through translation into a precisiation language. In the case of PNL, the precisiation language is the Generalized Constraint Language (GCL). A generic generalized constraint is represented as Z isr R, where Z is the constrained variable, R is the constraining relation and 1 is a discrete-valued indexing variable whose values define the ways in which R constrains Z. The principal types of constraints are: possibilistic (r=blank); veristic (r=v); probabilistic (r=p); random set (r=rs); usuality (r=u); fuzzy graph (r=fg); and Pawlak set (r=ps). The rationale for constructing a large variety of constraints is that conventional crisp constraints are incapable of representing the meaning of propositions expressed in a natural language—most of which are intrinsically imprecise—in a form that lends itself to computation.

The elements of GCL are composite generalized constraints that are formed from generic generalized constraints by combination, modification, and qualification. An example of a generalized constraint in GCL is ((Z isp R) and (Z, Y) is S) is unlikely.

By construction, the Generalized Constraint Language is maximally expressive. What this implies is that PNL is the largest subset of a natural language that admits precisiation. Informally, this implication serves as a basis for the conclusion that if a concept, X, cannot be defined in terms of PNL, then, in effect, it is undefinable or, synonymously, amorphic.

In this perspective, the highest level of definability hierarchy, which is the level above PNL-definability, is that of undefinability or amorphicity. A canonical example of amorphic concepts is that of causality. More specifically, it is not possible to construct a general definition of causality such that given any two events A and B and the question, “Did A cause B?” the question could be answered based on the definition. Equivalently, given any definition of causality, it will always be possible to construct examples to which the definition would not apply or yield counterintuitive results.

In dealing with an amorphic concept, X, what is possible—and what we generally do—is to restrict the domain of applicability of X to instances for which X is definable. For example, in the case of the concept of a summary, which is an amorphic concept, we could restrict the length, type, and other attributes of what we want to summarize. In this sense, an amorphic concept may be partially definable or, p-definable, for short. The concept of p-definability applies to all levels of the definability hierarchy.

In essence, PNL may be viewed as a collection of ordered pairs of the form (p, p^*), where p is a precisiable proposition in NL and p^* is a precisiation of p, that is, its translation in GCL. In this sense, PNL may be viewed as a dictionary in which p is an entry and p^* is its meaning.

In scientific theories, a concept, X, is almost always defined as a crisp (bivalent) concept, meaning that the denotation of X is a crisp set in its universe of discourse. In THD, a concept, X, is associated with a quintuple (X, U, QCS, DF(L), D(DF)) in which X is the concept; U is the space of objects to which X is applicable; QCS is the qualitative complexity scale associated with X; DF(L) is a definition of X in a language L; and D(DF) is the domain of DF, that is, the set of objects to which DF is applicable.

The concept of a qualitative complexity scale plays a key role in THD. Basically, the qualitative complexity scale, QCS, is a linear clustering, QCC_1, QCC_2, ..., QCC_m, of qualitative complexity classes of objects in U such that: (a) objects in QCC_i are roughly equally complex in relation to the definition, DF, of X; and (b) objects in QCC_{i+1} have higher complexity than those in Q_i. For example, if X is the concept of volume, then QCC_2 may be class of objects like trees; and QCC_5 may be the class of objects like clothing. Each language in the definability hierarchy is associated with a critical threshold on the qualitative complexity scale such that the language cannot be applied to classes above the critical threshold.

As the lowest member of the definability hierarchy, the C language has a low expressive power, with the consequence that the associated critical threshold is near the low end of the of the qualitative complexity scale. In particular, the C language cannot be used to define fuzzy concepts. Thus, its use to define concepts which, in reality, are fuzzy concepts, leads to counterintuitive conclusions. An example is the conventional C-language-based definition of stability. Since stability, in general, is a matter of degree, its definition as a crisp concept leads to paradoxes similar to the ancient Greek sorites paradox. To define stability as a fuzzy concept—which in reality it is—what is needed is PNL. The same applies to the concept of causality. Thus, causality can be defined as a crisp concept only for complexity classes which lie close to the low end of the qualitative complexity scale.

Another important point is that almost every concept has some degree of amorphicity, with a concept such as causality being amorphic to a high degree. But even such basic concepts as volume, density, edge, derivative and optimality have domains of amorphicity which are apparent in many real-world settings. What this implies is that many basic concepts may require redefinition in terms of PNL.

Does PNL have a significant role to play in devising metrics of performance for intelligent systems? This is an issue that is not addressed in the brief sketch of the theory of hierarchical definability. But I have no doubt that it will, since the concept of intelligence is much too complex to lend itself to analysis through the use of existing bivalent-logic-based methods.

PART I
RESEARCH PAPERS

Plenary Lecture 3 - 1

Robot Intelligence for Tunneling and Confined Space Search and Rescue

John G. Blitch
Director
Center for Robot Assisted Search and Rescue (CRASAR)
P.O. Box 270346, Littleton CO 80127
303.697.7607, {crasarhq@aol.com, blitchj@aol.com}

ABSTRACT

The horrific nature of structural collapse and natural disaster has unquestionably risen to the forefront of American perspective in the aftermath of recent terrorist attacks of 09/11/01. In responding to these events, heroic rescue personnel are routinely faced with a tremendously complex, hazardous, and often frustrating task environment that all too often leaves them without success in the presence of seemingly endless streams of carnage and body bags. The use of robots to assist in mitigating this challenge has been validated with a moderate degree of success by recent response efforts at the World Trade Center, but the tele-operated nature of the systems employed there did not fully exploit robot employment potential.

This paper presents the case for machine reasoning in the context of robot assisted search and rescue. A description of the USAR challenge is presented first in an effort to develop the reader's appreciation for complexities and challenges of the problem at hand. This is followed by a brief evolutionary review of mobile robot development and the motivation behind a shift toward portable platforms. A more detailed review of CRASAR's evolution and its pioneering actions in response to the World Trade Center attack is provided next with emphasis on platform shortcomings and the need for semi-autonomous control schemes. A concept for USAR (Urban Search and Rescue) oriented micro-tunneling is presented next as a new challenge in volumetric reasoning. The paper concludes with sequence of grand research challenges for roboticists interested in the USAR task domain.

Keywords: robot, robotics, confined space rescue, search and rescue, tunneling, micro-tunneling, artificial intelligence, intelligent systems

PART I
RESEARCH PAPERS

3M1 - Measuring Intelligence

Metrics, Schmetrics! How The Heck Do You Determine A UAV's Autonomy Anyway?

Bruce T. Clough, Technical Area Leader, Air Force Research Laboratory
Wright-Patterson AFB, OH, Bruce.Clough@wpafb.af.mil

ABSTRACT

The recently released DoD Unmanned Aerial Vehicles Roadmap [*9*] discusses advancements in UAV autonomy in terms of autonomous control levels (ACL). The ACL concept was pioneered by researchers in the Air Force Research Laboratory's Air Vehicles Directorate who are charged with developing autonomous air vehicles. In the process of developing intelligent autonomous agents for UAV control systems we were constantly challenged to "*tell us how autonomous a UAV is, and how do you think it can be measured...*" Usually we hand-waved away the argument and hoped the questioner will go away since this is a very subjective, and complicated, subject, but within the last year we've been directed to develop national intelligent autonomous UAV control metrics - an IQ test for the flyborgs, if you will. The ACL chart is the result. We've done this via intense discussions with other government labs and industry, and this paper covers the agreed metrics (an extension of the OODA - observe, orient, decide, and act - loop) as well as the precursors, "dead-ends", and out-and-out flops investigated to get there.

Keywords: autonomy metrics, machine intelligence metrics, UAV, autonomous control

1. Background

At top levels of the US Department of Defense an effort has been initiated to coordinate researchers across the Services and industry in meeting national goals in fixed-wing vehicle development. The Fixed-Wing Vehicle Initiative (FWV) has broad goals across numerous vehicle technologies. One of those areas is mission management of UAVs. Our broad goal is to develop the technology allowing UAVs to replace human piloted aircraft for any conceivable mission. This implies that we have to give UAVs some level of autonomy to accomplish the missions. One of the cornerstones of the FWV process is the establishment of metrics so one know that a goal is reached, but what metrics were available for measuring UAV autonomy? Our research, in conjunction with industry, determined that there was not any sort of metric as we desired. Thus we set out to define our own [Note 1].

But what characteristics should these metrics have? We decided that they needed to be:

- Easily visualized such that upper management could grasp the concepts in a couple of briefing slides.
- Broad enough to measure past, present and future autonomous system development.
- Have enough resolution to easily track impact of technological program investments.

So, they had to be simple, apply to a broad range of systems, and yet exhibit good resolution. Obviously a simple task, but first let's look at what it means to be autonomous.

2. Quick Difference Between Autonomous and Automatic (our definition)

Many people don't realize that there is a significant difference between the words autonomous and automatic. Many news and trade articles use these words interchangeably. Automatic means that a system will do exactly as programmed, it has no choice. Autonomous means that a system has a choice to make free of outside influence, i.e., an autonomous system has free will. For instance, let's compare functions of an automatic system (autopilot) and an autonomous guidance system:

- Autopilot: Stay on course chosen.
- Autonomous Guidance: Decide which course to take, then stay on it.

Example: a cruise missile is not autonomous, but automatic since all choices have been made prior to launch.

3. We Need To Measure Autonomy, Not Intelligence

For some reason people tend to equate autonomy to intelligence. Looking through the proceedings of the last NIST Intelligent Systems Workshop there are several papers which do this, and in fact, the entire conference sets the tone that "intelligence is autonomy" [*3*]. They are not the same. Many stupid things are quite autonomous (bacteria) and many very smart things are not (my 3 year old daughter seemingly most of the time). Intelligence (one of a myriad of definitions) is the capability of discovering knowledge and using it to do something. Autonomy is:

- the ability to generate one's own purposes without any instruction from outside (L. Fogel)
- having free will (B. Clough)

What we want to know is how well a UAV will do a task, or better yet, develop tasks to reach goals, when we're not around to do it for the UAV. We really don't care how intelligent it is, just that it does the job assigned. Therefore, intelligence measures tell us little. So, although we could talk about the Turing Test [*1*] and other intelligence metrics, that is not what we wanted.

4. Well, It Should Be Easy To Find Metrics, One Has The Web And Other Info Sources, Right?

Well, one would think so, but after an exhaustive one-month search involving the author, Air Force Research Laboratory Library Staff, and several other search organizations we found two. Two. Now if the goal was "find machine intelligence metrics" we would have been inundated with piles of paper. In addition to the aforementioned workshop, we would be looking through hundreds of publications and papers. Maybe it was a good thing we were looking for autonomy metrics!

We had thought that maybe, just maybe, the folks working distributed autonomous robotic systems had looked at this problem, but our questions to experts in that field revealed that they are just starting to ask those questions themselves [*4*]. The problem the researchers have in this area is that the metrics they are coming up with are task specific - they don't have general metrics quantitatively measuring higher-level characteristics of autonomous robot control architectures.

So what were the two that we found? Los Alamos National Laboratory's "Mobility, Acquisition, and Protection" space [*6*], and Draper Laboratory's "Three Dimensional Intelligence Space"[*7*]. The following is a short discussion of each [*Note 5*].

1. Los Alamos National Lab: Mobility, Acquisition, and Protection (MAP)

MAP comes from the lab of Mark Tilden, who develops simple robots based on analog circuits [*10*]. He needed a way to quantify the autonomous nature of his systems, and teamed up with LANL Physicist Brosl Hasslacher to develop the "Mobility, Acquisition, and Protection" space, or MAP for short. Figure 1 is a diagram of MAP from [*6*]

As one might expect from the name, this method uses mobility, acquisition, and protection to measure the ability of an autonomous system to survive in the world.

- *Mobility* relates to the capability of utilizing movement in the environment. M0 implies no motion abilities where as M3 can move in three dimensions, and M- means that external force must be used to move object.
- *Acquisition* relates to the ability to gather, store, and utilize energy. A0 implies zero energy consumption or delivery, A4 means planned tactics used to efficiently extract, store, and utilize external energy, while A- indicates object uses a non-replenishable energy store
- *Protection* indicates the capability to defend oneself. P- indicates one is physically more fragile than the environment while P1 means one executes flight/hide behaviors against hostile stimuli, and P3 demonstrates tactical fight/flight behaviors.

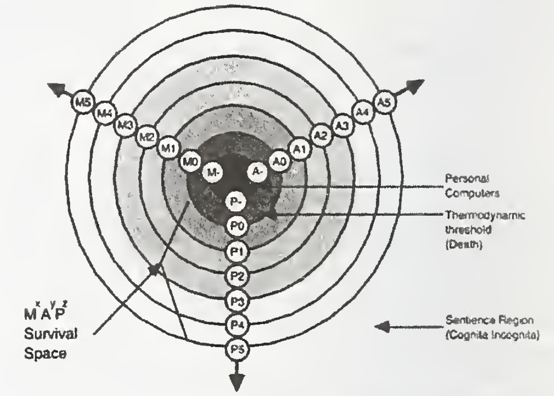


Figure 1: MAP Survival Space in which autonomous systems can be measured

These level metrics are fully described in Table 1 below. The space has three metrics has the three metrics outlines above, and six levels.

Level	Metric: Mobility	Metric: Acquisition	Metric: Protection
-	Motion Only Occure Under Application Of An External Force	Operates from a non-replentiehable energy source (battery, power line, etc.)	Negative Defensive capabilities (physically more fragile than the environment)
0	No Motion Abilities	Zero energy consumption or delivery	Zero defensive abilities (structural etrength equal to environment
1	Moves Deliberately In One Dimension	Can directly extract/apply external energy when available	Flight or hide behavior against hostile stimulus
2	Moves Deliberately In Two Dimensions	Can effciently extract/etore/utilize external energy	Fight or flight behavior against hostile stimulus
3	Moves Deliberately In Three Dimensions	Ueee focueed tactice to efficiently extract, etore and utilize external energy	Tactical fight/flight behavior against hostile stimulus
4	Capable Of dual-mode motion with tools, vehiclee, or application of specific deeign elements	Ueee planned tactice to efficiently extract, store and utilize external energy	Too, vehicle, or material use in fight/flight tactics
5	Human	Human	Human

Table 1: Level Descriptors For MAP Survival Space

MAP is actually quite a versatile visual tool, allowing disparate items to be plotted on the same page. Since there are three metrics, one can use a "radar chart" to display the measurements of a particular autonomous system, and this is excellent, since upper management likes radar charts! The Los Alamos researchers also realized this and included a MAP radar chart in their report. Showing this versatility, one can plot an ant, human, and a toaster on the same chart as is done in Figure 2! Tilden and Hasslacher successfully use MAP to illustrate the survival capabilities of the robots they design.

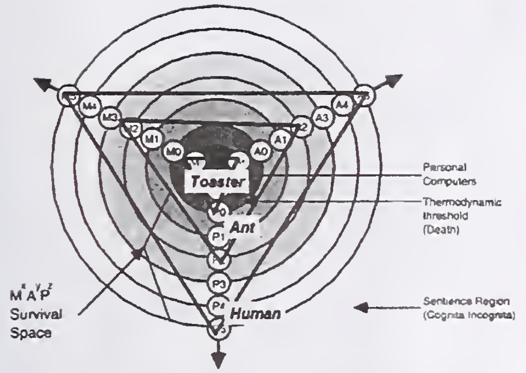


Figure 2: Various Objects plotted in MAP survival Space .

Can this be used to measure the autonomy of UAVs? Possibly. Figure 3 shows a plot of a multi-UAV neural net-based autonomous control system. This plot illustrates the limitations of MAP for our use. All UAVs would score M3 and A- - they move in three dimensions but require stored fuel. Protection ranges between P0 (structural strength to absorb damage) to P4 (groups of UAVs deliberately take out SAM sites). So, is this really useful for FWV autonomy measurement? No.

- Only one axis shows any variability, the others are fixed.
- The metrics just do not address operational characteristics of UAVs. They do not relate the autonomy present in the vehicle to the capability to perform useful missions.
- The metrics do not address interaction between UAVs (teams, swarms, etc.)
- The metrics do not allow us to adequately discriminate between different levels of autonomy. For instance, an RPV and an UAV with autonomy doing the same mission would score the same.

So although using MAP seems to make sense for simple robots, as a UAV autonomy measurement it isn't particularly useful.

So, the first metric space we examined could not fulfill our autonomous control system metric search, so we went on to investigate the other candidate we found – the "3D Intelligence Space" of Charles Stark Draper Laboratory.

2. *Draper 3D Intelligence Space*

Charles Stark Draper Laboratory (Cambridge, MA) has been developing robotic systems for military and other Federal customers for a number of years. They saw the same need to measure how well their systems could perform various tasks, and developed metrics under the sponsorship of the Office of Naval Research.

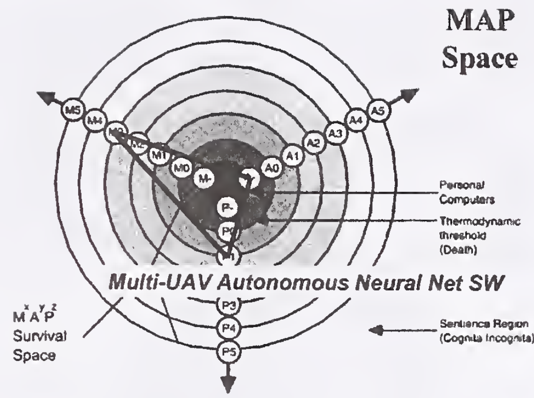


Figure 3: Autonomous Control System Plotted On MAP

These metrics were described in a paper [7] written by the Draper Lab researchers last year. This paper contained several different options to measure both intelligence and autonomy. Here we focus on the 3D Intelligence Space outlined in Table 2.

One can see this metric space has a couple desireable attributes:

- It has three metrics,so we can still use three-axis radar charts to represent the results, which will keep management happy(Figure CSD1).
- It has metrics which can be directly related to operational issues.

Level	Metric		
	Mobility Control	Task Planning	Situational Awareness
1	None, RPA Only	None, RPA Only	None, RPA only, or sensor as conduit
2	Operator Assisted	Waypoint or feature oriented	Low-level sensor processing , e.g. visual servoing (template tracking)
3	Get to waypoint, do one feature-based command	Interpret goals into action	Single-Sensor model matching
4	Integrate multiple actions	Multi-Agent Collaboration and C2	Integrated, multi-sensor fusion

Table 2: 3D Intelligence Space

Note that we made the distance between levels in Figure 4 increasing exponentially to represent the difficulties technicallly in going between steps.

We went ahead an plotted the same multi-UAV autonomous control systems used earlier in evaluation MAP space on the Draper radar chart. The results of this are in Figure 5. Note that this simple multi-UAV autonomous control system managed to "max-out" the metric space on all three axes, and highlight the fact that the resolution needs to be better. Other drawbacks include:

- *Task Planning* axis needs to be renamed. Many successful autonomous systems are based on pure reactive behaviors (such as insects). Task planning isn't a prerequisite to autonomy, it just allows better reactions to complicated situations.

- Situational awareness is based on the number of sensors and how they are fused, not on whether or not the autonomous system understands what's going on around it. In other words, this should be a measurement on how well the "big picture" is comprehended and understood.

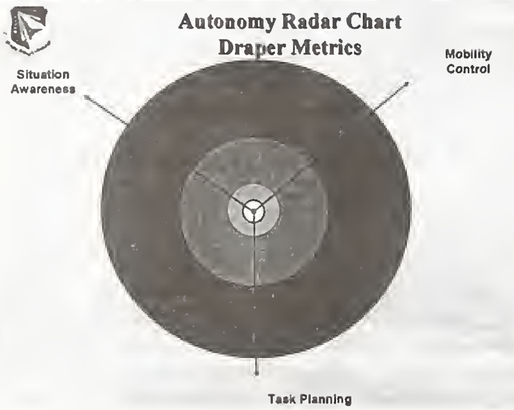


Figure 4: Radar Chart Of Draper Metrics

The drawbacks not withstanding, the Draper metrics provided us another good way of looking at the world.

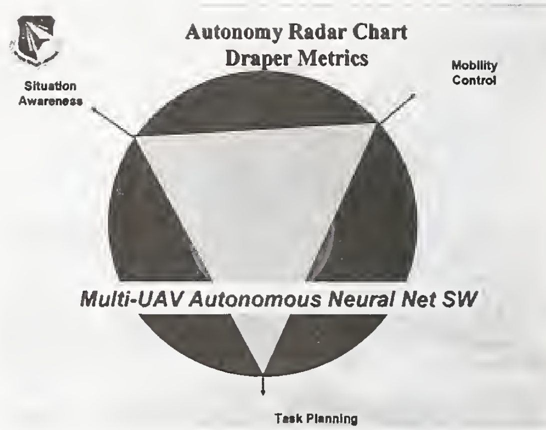


Figure 5: Autonomous Control System Plotted Using Draper Metrics

5. Initial Autonomous Control Level (ACL) Chart

We decided that since no existing metric space existed that could be directly used, we would integrate the best features of the ones we found with what we already used internally to represent where our technology was going. Table 3 is that first cut at an Autonomous Control Level (ACL) chart [Note 2].

Level	Level Descriptor	Perception/Situational Awareness	Analysis/Decision Making	Communication/Cooperation
10	Human-Like			
9	Multi-Vehicle Tactical Performance Optimization	Detaction & trecking of othar air vehicles within airspace	Full decision making cepability on-board Dynamicelly optimiza multi-ship group for tactical situation	Distributed cooparation with other air vehicles On-board deconfliction and collision avoidance Fully independent of supervision/control if desired; No centralized control within multi-UAV group
8	Multi-Vehicle Mission Performance Optimization	Detection & tracking of other air vehicles within local airspace OK to operate in controlled airspace w/o extemal control	Continuous mission/trajectory aveluation & replan - optimiza for currant mission situation Avoid collisions and replan/optimize trajectory to meet goals, atc	External suparvision - abort/recall or new overall goal On-board deconfliction & collision avoidance Distributed cooparation with other A/V's
7	Real-Time Multi-Vehicle Cooperation	Detection of other A/V's in local airspace Multi-threat detection/analysis on-board	Continuous flight path evaluation & raplen Compensate for anticipated system malfunctions, weather, atc - optimize trajectory to meet goals, managa resources, avoid threats, etc	On-board collision avoidance Uses off-board data sources for deconfliction & tracking Hierarchical cooperation with other A/V's
6	Real-Time Multi-Vehicle Coordination	Detection of other A/V's in local airspace Single threat detection/analysis on-board	Event-drivan on-board, RT flight path replen - goal driven & avoid threats RT Haalth Diagnosis; Ability to compansate for most failures end flight conditions - Inner loop changes reflected in outer loop performance	On-board collision evoidance Uses off-board data sources for deconfliction & tracking Assumed acceptance of replen; External supervision - rejection of plan is axception Possible close air space separation (1-100 yds)
5	Fault/Event Adaptive Vehicle	Automated Aerial Refueling & Formation sensing Situational ewareness supplemented by off-board data (threats, other A/Vs, etc)	Event-driven on-boerd, RT trej replen to new destination RT Health Diagnosia; Ability to compansate for most failures and flight conditions; Ability to predict onset of failures (a.g. Prognostic Health Mgmt) On-board assessment of status vs trajectory	On-board darived vehicla trajectory "corridors" Uses off-board date sources for deconfliction & tracking External supervision - accept/reject of raplan Possible close air space seperation (1-100 yds) for AAR, formation in non-thraat conditions
4	Robust Response to Anticipated Faults/Events	Threat sensing on-board	RT Heelth Diagnosis (Can I continue with these problems?); Ability to compensate for most feilures end flight conditions (e.g. Adaptive inner loop control); Automatic trajectory axecution; On-board assessment of status vs mission completion	Secure, within LOS electronic tether to nearby friendlies Offboard darived vehicla "corridors"; Medium vehicle airspace separation (100's of yds) Threat analysis off-board
3	Limited Response to Real Time Faults/Events		RT Health Diag (What is the extent of the problems?) Ability to compansata for limited failures (a.g. Reconfigurable Control) Autometic trajectory execution	Health Status monitored by axternal suparvision Off-board replan; Waypoint plen upload Wide airspace separation requirements (miles)
2	Pre-loaded Alternative Plans		RT Health diagnosis (Do I have problems?) Automatic trajectory execution (vie waypoints) Preloaded elternative plens (e.g. ebort)	External commands - eltemativa plans, epprovals, aborts Reports status on request or on schedule Wida airspace separation requirements (miles)
1	Execute Preplanned Mission	Situational ewareness via Remote Operator Flight Control and Navigation Sensing	Robotic/Preprogremmed Pre/Post Flight BIT	External control vie low lavel commands Reports status on request Wide airspace separation requirements (miles) No on-board knowledge of other air vehicles - all actions ere preplanned
0	Remotely Piloted Vehicle	Flight Control (attitude, rates) sensing Nose camera Situational ewareness via Remote Pilot	N/A	Remotely Piloted Vehicle status date via telemetry

Table 3: Initial ACL Metrics Chart

We kept three metrics since we liked the idea of representing systems as areas on a radar chart when briefing management. We added ten levels for better resolution between remotely piloted aircraft and fully autonomous UAVs. The metrics related to operational issues while still being attached somewhat to technological systems. Populating the levels was a group endeavor, with a team of researchers, program managers, tech area leads, and contractor experts determining the meaning of each level: "1" was simple – the traditional remotely piloted aircraft (RC-type) while "10" was "human". The trick was populating the eight levels between. As with the Draper metric space in Figure 4 we represented the radar chart as having levels which are exponentially more difficult. We then plotted the same multi-UAV autonomous control system as before, and the result is in Figure 6. We recovered some of the resolution lost in the Draper metrics; however, we still had some issues with the metrics:

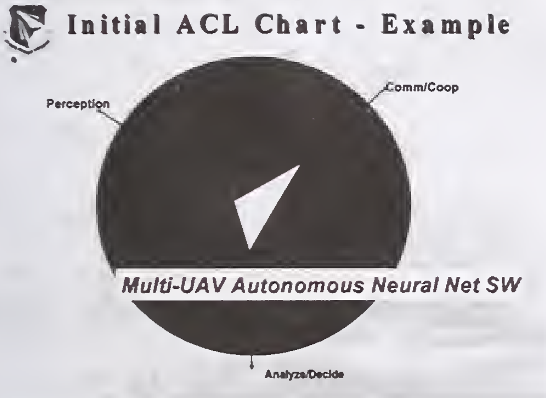


Figure 6: Autonomous Control System Plotted On Initial ACL Radar Chart

- The metrics weren't broad enough to cover UAVs acting on strategic knowledge to achieve strategic results. Our chart limited them to tactical thinking only.
- One metric mixed cooperation with communication – mixing a "what" with a "how".

In general, we tried to stuff as much into three metrics as possible to retain simplicity in briefing presentation and in the process lost the capability to split out issues of multi-UAV control and human-UAV interaction, to name two.

We were going to press ahead and use this metric space when one of the autonomous control system development engineers came up with a good idea [Note 3].

6. If You're Replacing A Human, Why Not Measure Like One?

The great insight was this: we are designing algorithms, agents if you will, to replace pilot decision functions. Machines replace human – so why not look to the human effectiveness community for metrics? Modify the OODA (observe, orient, decide & act) loop - originally developed to show how to get inside your enemy's decision loops - [8, & Note 4] for our use, and populate the levels with modifications of the qualifiers of the initial ACL chart. Table 4 is what we developed using this insight. The same team of experts that developed the initial ACL chart also worked on the new ACL chart to ensure consistency with earlier thoughts. We lost the three axes representation, which means we lost the ability to generate the "simple" radar charts which makes management happy; however, we gained better resolution between metrics which, at least in our thoughts, more than made up for that.

Since we have developed the ACL chart, we've used it to both assess the current UAV efforts, and to extract from that where our own technical efforts must go. Nationally, we've developed time-phased autonomous system goals to put our autonomous systems roadmap together. The ACL has been published as part of the DoD UAV Roadmap [9], and other DoD Labs use it to measure their autonomy development. Locally we've developed technical area roadmaps putting programs together to meet the time-phased ACL goals. The ACL chart also acts as a program advocacy tool, allowing us to show management how each program fits into increasing ACL capability for each metric, and also how each program investment integrates into the national strategy. Our experience from using it for one budget planning cycle has been very positive:

- Once management was briefed on the chart and it's development (and some in management had ownership in it's development) it was accepted as the tool to measure program goals.
- It provided clear indications of where the technology was targeted and what national goal it met, allowing better informed budget planning decisions.
- We have common ground for talking amongst other Federal technology development organizations, universities, and industry. Each of us has a much clearer picture as to where technological programs fit.

7. Summary

Our work with autonomous UAVs indicated to us that we needed metrics to measure the progress of our programs building that autonomy. The same issues existed on a national level, so we decided to develop metrics for the national-level effort, then apply those to our local program planning process.

Our literature search for autonomous system metrics only returned two references for metrics. Both we examined and used on an example problem. Although each wasn't directly applicable, concepts of each, integrated with our own existing ideas, formed an initial ACL chart. This chart was modified based on concepts human dynamists had developed - specifically the OODA loop. The resulting set of metrics captured our original intent. [*Note 4*]

Since development, the ACL metrics have been used successfully at the Air Force Research Laboratory in developing plans and programs in autonomous UAV controls research. The ACL chart is in current review at DOD levels to be applied across the services as part of the FVW initiative. With this development we are pressing ahead in the assessment of possible sub-metrics to better hone our program planning.

8. References

1. Turing, A. M. "Computing Machinery And Intelligence", Journal of the Mind Association, Vol. LIX, No. 236. Oxford Press. UK. October 1950.
2. Clough, B. *Autonomous UAV Control System Safety – What Should It Be, How Do We Reach It, And What Should We Call It?* NAECON 2000 paper. October 2000
3. A. Meystal et al., "Measuring Performance of Systems with Autonomy: Metrics for Intelligence of Constructed Systems," NIST Workshop, "Performance Metrics for Intelligent Systems," August 14-16, 2000
4. Parker, L.E. "Evaluating Success in Autonomous Multi-Robot Teams: Experiences from ALLIANCE Architecture Implementations". Unpublished (as of the time of this writing) paper. Center for Engineering Science Advanced Research, Oak Ridge National Laboratory. Fall 2000.
5. "Measuring the Performance of Intelligent Systems" Web publication available from http://www.isd.cme.nist.gov/conferences/performance_metrics/Measuring.html. Intelligent Systems Division, National Institute for Science and Technology, Spring 2000
6. Hasslacher, B., Tilden, M. W. "Living Machines" Los Alamos National Laboratory. 1995.
7. Cleary M., Abramson M., Adams M.B., Kolitz S. "Metrics For Embedded Collaborative Intelligent Systems" Charles Stark Draper Laboratory, Inc. 2000
8. Boyd, J.R. *Discourse on Winning and Losing*. Found on the web at http://www.defense-and-society.org/FCS_Folder/boyd.htm
9. *Unmanned Aerial Vehicles Roadmap 2000-2025*. Office Of The Secretary Of Defense, Washington DC. April 2001
10. Trachtman, P. "Redefining Robots". *Smithsonian Magazine*. Smithsonian, Washington D.C. February 2000.
11. Sheridan, T., Parasuramam, R. "A Model for Types and Levels of Human Interaction with Automation", *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, VOL. 30, No. 3. IEEE May 2000.

9. Notes

1. Not that we didn't have any. We had already split autonomy into four levels depending on the amount of human interaction and where it occurred. These are:

- Remotely Piloted: The UAV is simply a remotely piloted aircraft with the human operator making all decisions.
- Remotely Operated: The human allows the UAV to do the piloting, but outer loop decisions are made by the human (like where to go and what to do once there). The UAV is a "mother-may-I" system, asking the human permission to do tasks.
- Remotely Supervised: The human allows the UAV to execute its own tasks, only taking command if the UAV fails to properly execute them.
- Fully Autonomous: The UAV receives goals from the humans and translates that into tasks which it does without human intervention. The UAV has authority to make all decisions.

2. Most of the grunt work in putting the chart together was done by Dan Thompson, AFRL/VACC, and Dr. Alan Burkhard, AFRL/VAC - the rest of us got to snipe at it.
3. The researcher's name is Bob Smith, and besides coming up with decent ideas he also has developed a formation flight agent for UAV formations which uses a blend of deliberate and emergent behavior.
4. Boyd's OODA loop, originally developed to illustrate how to take advantage of an enemy, has been grasped wholeheartedly by business management folks. The observe, assess, design, and act (OADA) loop organizational dynamists use to explain how decisions are made is a direct descendant.
5. I know, you're wondering about Sheridan's Autonomy Levels. Truth of the matter is that if you search for "autonomy", "metrics", "measuring autonomy", etc., you don't run into Sheridan. Had I searched using "teleoperation" I would have found the Sheridan Autonomy Levels [*11*].

Level	Level Descriptor	Observe	Orient	Decide	Act
		Perception/Situational Awareness	Analysis/Coordination	Decision Making	Capability
10	Fully Autonomous	Cognizant of all within Battlespace	Coordinates as necessary	Capable of total independence	Requires little guidance to do job
		Battlespace inference - Intent of self and others	Strategic group goals assigned	Distributed tactical group planning	Group accomplishment of strategic goal with
9	Battlespace	(allies and foes).		Individual determination of tactical goal	no supervisory assistance
	Swarm	Complex/Intense environment - on-board tracking	Enemy strategy inferred	Individual task planning/execution	
	Cognizance			Choose tactical targets	
8		Proximity inference - Intent of self and others	Strategic group goals assigned	Coordinated tactical group planning	Group accomplishment of strategic goal with
	Battlespace	(allies and foes)		Individual task planning/execution	minimal supervisory assistance
	Cognizance	Reduced dependence upon off-board data	Enemy tactics inferred	Choose targets of opportunity	(example: go SCUD hunting)
			ATR		
7		Short track awareness - History and predictive battlesp	Tactical group goals assigned	Individual task planning/execution to meet goals	Group accomplishment of tactical goal with
	Battlespace	data in limited range, timeframe, and numbers	Enemy trajectory estimated		minimal supervisory assistance
	Knowledge	Limited inference supplemented by off-board data			
6	Real Time		Tactical group goals assigned	Coordinated trajectory planning and execution to meet	Group accomplishment of tactical goal with
	Multi-Vehicle	Ranged awareness - on-board sensing for long range,	Enemy location sensed/estimated	goals - group optimization	minimal supervisory assistance
	Cooperation	supplemented by off-board data			
					Possible close air space separation (1-100 yds)
5	Real Time	Sensed awareness - Local sensors to detect others,	Tactical group plan assigned	On-board trajectory replanning - optimizes for	Group accomplishment of tactical plan as externally
	Multi-Vehicle	Fused with off-board data	RT Health Diagnosis; Ability to compensate for most	current and predictive conditions	assigned
	Coordination		failures and flight conditions; Ability to predict onset of	Collision avoidance	Air collision avoidance
			failures (e.g. Prognostic Health Mgmt)		Possible close air space separation (1-100 yds) for
			Group diagnosis and resource management		AAR, formation in non-threat conditions
4		Deliberate awareness - allies communicate data	Tactical plan assigned	On-board trajectory replanning - event driven	Self accomplishment of tactical plan as externally
	Fault/Event		Assigned Rules of Engagement	Self resource management	assigned
	Adaptive		RT Health Diagnosis; Ability to compensate for most	Deconfliction	
	Vehicle		failures and flight conditions - inner loop changes		
			reflected in outer loop performance		Medium vehicle airspace separation (100's of yds)
3	Robust Response	Health/status history & models	Tactical plan assigned	Evaluate status vs required mission capabilities	Self accomplishment of tactical plan as externally
	to Real Time		RT Health Diag (What is the extent of the problems?)	Abort/RTB if insufficient	assigned
	Faults/Events		Ability to compensate for most control failures and		
			flight conditions (i.e. adaptive inner-loop control)		
2		Health/status sensors	RT Health diagnosis (Do I have problems?)	Execute preprogrammed or uploaded plans	Self accomplishment of tactical plan as externally
	Changeable		Off-board replan (as required)	in response to mission and health conditions	assigned
	Mission				
1	Execute	Preloaded mission data			
	Preplanned	Flight Control and Navigation Sensing	Pre/Post Flight BIT	Preprogrammed mission and abort plans	Wide airspace separation requirements (miles)
	Mission		Report status		
0	Remotely	Flight Control (attitude, rates) sensing	Telemetered data	N/A	Control by remote pilot
	Piloted	Nose camera	Remote pilot commands		
	Vehicle				

Table 4: Final ACL Chart

Towards Measuring the Performance of Architectural Components of Autonomous Vehicular Systems

Craig Schlenoff, Lawrence A. Welsch, Raj Madhavan, Noah Zimmerman
National Institute of Standards and Technology
craig.schlenoff, lwelsch, raj.madhavan, noah.zimmerman @nist.gov

Abstract

For a vehicular system to act "intelligent", the system must be able to 1) sense in a dynamic domain; 2) model the domain internally; 3) determine possible courses of action to accomplish a goal in the domain; and 4) be able to assess the various courses of actions to determine which is best. The actions that the system ultimately performs are a function of all of these components. Solely assigning performance metrics to the resultant action of the intelligent system does not evaluate any one of these components individually, and therefore leaves some doubt as to how to measure what each component contributes to the overall behavior of the system. Thus we are not looking at a single number, but a matrix of numbers that characterize the performance of the system.

In this paper, we are exploring a mechanism to assign performance metrics to the part of the system that models the domain internally, the internal knowledge representation of intelligent vehicular systems. We do not consider that part of a system that translates the raw sensory input from a vehicle's sensors to other representations. Rather we simulate a predefined set of sensory inputs, and evaluate the resulting knowledge representation based.

1 Introduction

Darwin was the first to propose the importance of natural intelligence for biological entities. He suggests that intelligence is the result of billions of years of natural selection, emerging from a competitive struggle for survival [1]. Measuring the *intelligence* of intelligent systems presents several challenges. A universal scalar value of intelligence is difficult to ascertain in a machine due to the restrictive nature of most domains.

Additionally, it is more difficult to make judgments based on the relative success of particular behaviors. However, in machines we have the advantage of being able to monitor the internal states. This enables us to make more accurate deductions about 1) the methods employed by the system to complete the task, and 2) the intermediate states that it traversed. The system can then be evaluated based on a relationship between the complexity and efficiency of the method and the precision of the final state.

There have been attempts to provide qualitative and quantitative measure to knowledge representations [10], though not, until recently, have they been applied to measuring the internal knowledge representations within autonomous vehicular systems. Gruninger and Fox have applied the concept of competency questions to formal ontologies to test their ability to answer the questions they were designed for [8]. McGuinness et al. have also explored approaches to testing the content of ontologies after multiple ontologies are merged by using a tool called Chimaera [9]. More recently work has been done to develop tests for text retrieval systems [11] and autonomous vehicle systems [12]. Research has also been done considering the performance of rule chaining in generic expert systems [13]. In this paper we are considering how to best take advantage of Real-Time Control System[1] architecture (described below) to measure the performance of the architectural components that contribute to the vehicle's behavior.

For a vehicular system to act in an intelligent manner, the system must be able to 1) sense in a dynamic domain; 2) model the domain internally; 3) determine possible courses of actions to accomplish a goal in the domain; and 4) be able to assess the various courses of actions to determine which is best. The actions that the system ultimately performs are a function of all of these components. Solely assigning performance metrics to the resultant action of the intelligent system does not evaluate any one of these components individually, and therefore leaves some doubt as to how to measure what each component contributes to the overall behavior of the system.

We have selected the Real-Time Control System (RCS)[1] as the architecture for evaluating intelligent systems. RCS is a hierarchical distributed real-time control system architecture that allows for modular and device independent algorithms to be developed for intelligent systems. A node in the RCS reference model architecture is shown in Figure 1.

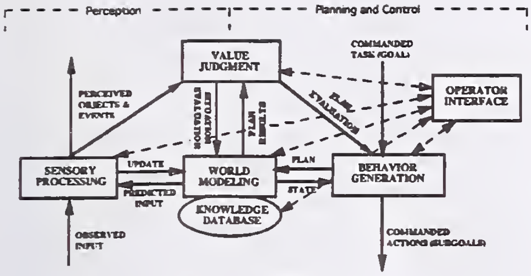


Figure 1: RCS Node

The functional elements of an intelligent system can be broadly considered to include: behavior generation (task decomposition and control), sensory processing (filtering, detection, recognition, grouping), world modeling (store and retrieve knowledge and predict future states), and value judgment (compute cost, benefit, importance, and uncertainty). These are supported by a knowledge database (KD), and a communication system that interconnects the functional models and the knowledge database. This collection of modules and their interconnections make up a generic node in the RCS reference model architecture. Each module in the node may have an operator interface.

Though several contemporary architectures exist in the literature for designing intelligent systems, our motivation for selecting RCS is many fold:

- In the last fifteen years, behaviorist architectures [2] [3] have gained popularity for their ease of implementation. However, within such architectures, long-term planning is not possible since only a single behavior can be selected for execution. Other disadvantages include the inability to fuse sensor data to arrive at a single best estimate of the state of the world (in some probabilistic sense) and the lack of internal representation of the world.
- RCS is a proven architecture with more than 200 person-years of research and development in intelligent control theory. It has been implemented and tested thoroughly both in the industry and academia in different operating domains under varying operating conditions. For example, RCS has been implemented as the reference model architecture for the design, engineering, integration, and testing of eXperimental Unmanned Vehicles for the DoD Demo III program [1] [4].
- RCS is supported in terms of software and updates and thus it constantly evolves through a number of versions at National Institute of Standards and Technology (NIST) and elsewhere [5]. For additional advantages, see pp. 128 of [6].

For the purpose of this paper, we are exploring a mechanism to assign performance metrics to the part of the system that models the domain internally, the internal knowledge representation of intelligent vehicular systems. We hold the sensory component constant and do not consider the behavior and value judgment components. In other words, we simulate a pre-defined set of sensory inputs, and evaluate the knowledge representation based on those sensory inputs. There would be no actions physically performed, nor would there be any value judgment implemented. In this paper, we explore developing a test harness for autonomous systems, focusing on each combination of knowledge representation components and functions. Thus, the test harness can be seen as a matrix, with the components along one axis and the functions along the other, and each cell composed of a series of questions testing the knowledge representation's ability to provide the stated function using the pertinent component, if appropriate. For example, a question such as "Where do you expect a given moving object to be at time=10?" may be appropriate to test the intelligent system's "prediction" function using its "inferencing" and "knowledge being represented" components.

In Section 2, we discuss a test harness, including the data flow through the harness and the places in the RCS hierarchy that would be appropriate to test. Section 3 discusses the typical purposes/functions of a world model. Section 4 describes the components of any knowledge representation, and discusses pertinent questions that could be asked to test those components of the knowledge representation. Section 5 brings the previous two sections together into a matrix, and discusses future work that should be done to address the development of the proposed test harness.

2 The Test Harness

2.1 Data Flow

The goal of this work is to test the world modeling capabilities of an autonomous vehicular system without requiring the system to be physically relocated to a test site, nor to require that the system have to perform any physical behaviors. The system's world modeling capabilities would be tested by a series of questions and answers, where the answers to the questions would be assigned a score based upon a series of performance

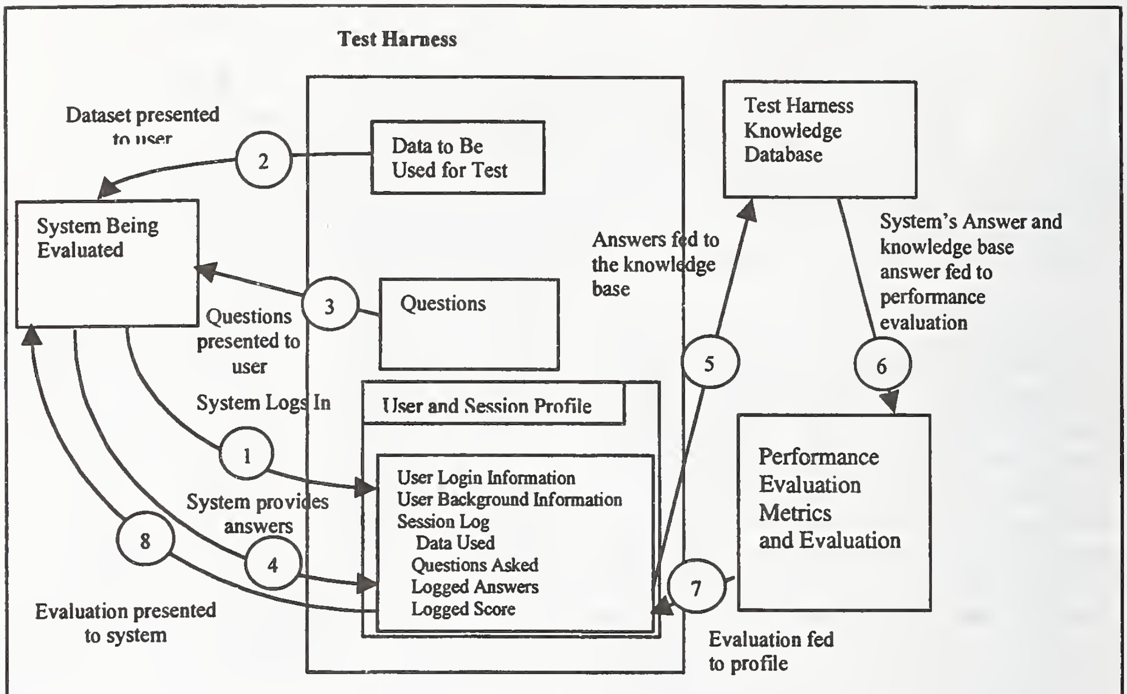


Figure 2: Data Flow in the Test Harness

evaluation metrics. Figure 2, along with the supporting text, shows the data flow pertaining to the interaction a system would have with the test harness, and is described in detail.

Figure 2 contains three main components: the system being evaluated, the test harness, and the knowledge base / performance evaluation components. The test starts when the 'system being evaluated' first registers by entering in its ID and password (number 1 in Figure 2). At this point, the user can choose between a series of sample sensory data to use for the test, sorted and rated by its level of difficulty (to be discussed in a future paper) (2). The system then has a predetermined amount of time to receive and process the data. After the data is processed, a series of questions that correspond to that data set are posed to the user (discussed in Sections 3 and 4) (3). These questions may also be rated by their level of difficulty. The user responds to these questions by providing an answer, as well as a description of how that answer was determined (4). This information, along with the amount of time that was taken to determine the answer, is noted in the user's profile. This information is the passed to the test harness knowledge base where it is compared with system's knowledge base's response to the same questions (5).

The answers from the systems and the knowledge base are then passed to the evaluation component, where predetermined metrics are used to assign a score to the system's answer (6). The score would be a function based upon the "correctness" of the answer (e.g., the answer was two, but the system thought the answer was four), the procedure used to come up with the answer (e.g., what were the equations used and the assumptions made when the answer was being determined), the amount of time it took to produce the answer, and the amount of detail provided in the answer (e.g., the answer was two, but the system responded with an answer of "between one and five"). This score is then fed back to the system's profile to be logged (7), and reported to the system (8).

There are many interesting and challenging research areas within the scope of this framework, including the types of sensor data to be presented to the user, the types of questions that should be asked to the user in response to the sensor data, the information to store in the knowledge base to evaluate the answers the system provides, the appropriate evaluation metrics to use in evaluating the answers (including the weights to put on each of the factors described in the previous paragraph), the details of the communication specifications between the system and the test harness, the interfaces and the representation of the information to be passed between the various internal components of the framework, as well as the mechanism to allow a system to supply an explanation of how an answer was produced. This paper focuses solely on the questions that are asked on the system being evaluated. Future papers will focus on the other challenges mentioned above.

2.2 *Applying the Test Harness to Various Components in RCS*

The test harness described above is generic and may be used to test an entire node in the RCS hierarchy (as shown in Figure 1) or just a component of a node. If the entire node is being tested, then raw sensory data would be fed to the "systems being evaluated" as input (as indicated by the bottom left arrow entering the box) and the output plan of the RCS node (as indicated by the bottom right arrow exiting the box) would be evaluated.

Instead of looking at the entire RCS node, one could only test one or more components of the node, thus focusing the attention on only a small subset of the node. In this paper, we are interested in the contribution of the World Model / Knowledge Database component as shown in Figure 3 below. In this case, we would be feeding processed sensory data to the world model (thus the sensory processing is not considered), and can query the world model about what it perceives, where is expects objects to be in the future, etc. (thus the planning is not considered since the world model is never asked to generate a plan, it is just asked to answer questions about what it is presented).

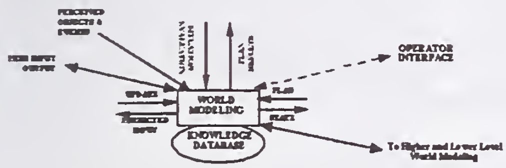


Figure 3: World Model and Knowledge Database

Although this paper solely focuses on applying the proposed test harness to systems based upon the RCS architecture, there is nothing in the design of the test harness that precludes it from being applied to other systems. The only assumption that this test harness design makes is that there is a clear place in the "system being evaluated" to which information can be fed, that there is a clear place in the "system being evaluated" from which information can be read, and there is an appropriate set of questions and evaluation metrics which can be applied to evaluate the system.

In the next section of the paper, the functionality of the test harness is exposed and test interactions proposed.

3 Functions of a World Model

The world model can be thought of as a component of the brain of the intelligent system. Just as the brain contains a representation of the environment, the world model contains a representation of its surroundings, and as such, must be able to use that representation to the benefit of the system that is immersed in that environment. The world model must inform the intelligent system on the potential results of action, similar to the way the brain informs the human body of the possible consequences of actions.

The world model can be thought to be comprised of four functions: maintenance and updating of the knowledge base, prediction of sensory input, response to queries for information required by other processes, and simulation. This is described in detail in [6]. In this section of the paper, we will provide examples of the types of queries that the world model would be expected to answer to perform these functions.

3.1 *Maintenance and Updating of the Knowledge Database*

The world model in its entirety is the intelligent system's best estimate of the world at the given time. The world model can be thought of as comprising a number of knowledge databases, where each knowledge base is a store of information about the world.

To ensure that the representation of the world is up-to-date, the world model must constantly be updated as new information is available. Examples of ways that the world model could be updated include:

1. As new processed sensor data is available and entities are identified, the world model must compare the actual location of the sensed images to the location in which the world model predicted that it would be. (What is the difference between the actual location of entity A and the predicted location of entity A?, How can the current prediction parameters be changed to provide more accurate predictions?)
2. As time elapses, information will move from immediate experience to short term memory, to long term memory. The world model must seamlessly allow for the migration of information into these parts of the world model, as well as transform the representation

of this information between different representation approaches. (What information should be moved to short-term memory? To long-term memory?)

3. As time elapses, new entities will appear in the intelligent system's environment, and some entities will no longer exist. The world model must be able to introduce these new entities into the knowledge database, determine which ones are most important to track, and delete those entities that no longer exist or are no longer of interest. (What new entities exist that were not previously modeled in the knowledge base? Which of these entities are important to track? What are the pertinent characteristics of those entities? What are the criteria for deleting entities from the knowledge base?)
4. In the real world, relationships exist between entities, events, and situations. It is important to maintain these relationships within the world model. (What are the important relationships in a given environment?, How should those relationships be represented?, For what time extent do those relationships hold?)

3.2 Prediction of Sensory Input

In addition to capturing the data that is passed to it by the sensors, the world model must also predict where it believes the next set of sensed data will be. Being able to accurately predict where an object is expected to be at a time in the future is essential for areas such as image processing, path planning, and collision avoidance. Accurate prediction algorithms allow the world model to better predict where an object is expected to be at some time in the future, along with a stated degree of uncertainty, and therefore make plans that account for that predicted future location.

Questions that may be asked within this function of the world model include "What is the predicted location of entity A given data pertaining to its previous location?", "What are the appropriate algorithms to provide the prediction?", "What are the criteria for updating the prediction parameters?")

3.3 Response to Queries for Information by Other Processes

The world model is the primary source for information within the intelligent system. It is designed to be an information repository, and as such, must interface with other components of the hierarchy that have a need to retrieve information from it, whether explicitly or implicitly represented. More specifically, the world model provides the following functions:

1. The world model responds to requests from the sensory processing, behavior generation, and operator interface components of the hierarchy. The sensory processing component may ask for the predicted attributes and states of an entity. The behavior generation component may request the predicted identity of entities in the environment, as well as characteristics of those entities (e.g., if the entity was a car, how fast is the car going? In what direction? What is the fastest the car can go?, etc.). The operator input may ask for the state of the intelligent system at the current time. (What is the predicted location, speed, orientation of entity A at time = t+1?, What is the object perceived by the sensors, and what are the pertinent characteristics of it?)
2. The world model performs coordinate transformations, when necessary, and accounts for the motion of the sensor platforms that affect sensor input.
3. The world model deduces additional information from the knowledge database that is not explicitly represented, but can be deduced from the information that is represented. (Given the information known about an object, what additional information can I infer about the entity that is not explicitly represented?)

3.4 Simulation

In almost any application, it is useful to simulate the results of an action before the action is physically performed. More specifically, the simulation aspect of the world model provides the following functions:

1. The world model uses the knowledge in the knowledge databases to simulate the results of possible plans generated by the behavior generation module.
2. The world model can compute all of the sets of actions which can be performed to produce a desired output.
3. The world model interfaces with the value judgment component to evaluate the cost/benefit of the proposed action based on the simulation (What are the appropriate cost algorithms?, Given a cost algorithm, what is

which plan provides the most benefit at the least cost?).

4 Knowledge Representation Measurements

The previous section described functions that the world model within an intelligent system is expected to perform. Based on those functions we posed queries that the world model are needed to support the functions. This section proposes measures for the knowledge database within the world model. By considering each measure against each query, we derive the matrix described in the conclusions.

The knowledge database can be thought of as having three attributes: 1) the formalisms for representing knowledge (i.e., how the knowledge is captured), 2) the actual knowledge the system has represented at any given time (e.g., the data that is captured within the knowledge database), and 3) the mechanism(s) available for accessing, querying, and inferencing over the represented knowledge. Each of these attributes provides a different set of measures, for the value that each brings to the overall world model.

4.1 Measuring the formalisms for representing knowledge

A KD may contain a variety of different types of formalisms for representing knowledge in its database. For example the KD may contain formalisms to represent:

- Raw sensory data collected directly from sensors;
- Map and/or geometric data where map data might provide coordinates for landmarks, roads, and topological features.
- Symbolic and/or rule data that might contain rules such as *drive on the right side of the road,* or *enter buildings through an opening*; and
- Links or associations between the different types of data.

When determining the metrics for measuring the formalisms for representing knowledge, one may consider the following criteria:

1. The number of different types of representations that the KD supports;
2. The complexity level the formalism can support. For example, in the case of symbolic representation, is the representation capable of representing Boolean algebra, first order predicate calculus, etc.;
3. The detail or granularity in which the fundamental physical units may be represented;
4. The size of the largest set that be represented – finite, countable, etc.; and
5. The number of mechanisms in which one can group knowledge.

Each measure can be considered for each question described in section 3. For example, for a particular query, the measure would be the number different types of representations of data that were involved in generating a response to the query.

4.2 Measuring the actual representation of the knowledge

At any given instant in time the world model has a set of information that is captured within its knowledge databases. One can measure the captured knowledge using the following types of metrics:

1. The quantity of different contexts/concepts[1] that are represented;
2. The quantity of contradicting knowledge, possibly organized by contexts;
3. The scale of complexity [7] of the most/least complex concept represented (not the complexity of the formalism, but rather the concept itself);
4. The numbers of links among concepts; and
6. The depth of the hierarchy tree (e.g., how many "levels" are in the representation?).

Again each measure would be considered against the each query described in section 3.

4.3 Measuring the mechanisms for accessing in and inferencing over the knowledge database

Finally we need to evaluate the performance of the mechanisms that respond to requests of the KD (the inference or query mechanism). The measures considered are:

[1] By concept/context is meant a collection of knowledge that is not self-contradictory. Frequently a context/concept is a way of organizing knowledge so as to make the knowledge easier to find.

1. The length of time[2] the system takes to find a particular fact, rule, assertion already in the KD.
2. The minimal time to combine a fact with an assertion;
3. The speed to switch representation formalisms (with and without links);
4. The minimum time to combine knowledge in one representation formalism with another; and
5. The quantity of different inferencing mechanisms that exist.

Again, each measure would be applied to each query. For example for the query *What is the difference between the actual location of entity A and the predicted location of entity A?*, the first measure would be, the minimal time to retrieve a fact necessary to addressing the query.

		Knowledge Representation Characteristics		
World Model Functions		formalisms for representing knowledge	knowledge being represented	mechanisms for information access and inferencing
	maintenance and updating			
	prediction of sensory input			
	response to queries for information required by other processes			
	simulation			

Table 1: Test Matrix

5 Conclusions / Future Work

In this paper, a test harness was introduced with an emphasis on the types of questions that would be needed to test the world modeling capabilities of an intelligent system. One can imagine a series of questions that would test certain expected functions of the autonomous system's world model, with respect to specific characteristics of the knowledge representation such as the way the knowledge is represented, the exact knowledge that is represented, and the mechanisms for querying that knowledge. These questions would logically fall into the matrix, as shown in Table 1, with specific questions tailored for each cell in the matrix.

Work has recently been started on implementing the framework of the test harness, using an agent-based infrastructure, in a web-based environment, such that the interaction with the test harness would be web-based calls with a web server located at NIST. However, much work remains to be completed.

As mentioned in Section 2, there are many interesting and challenging research areas within the scope of this test harness that have yet to be addressed, including:

- the types of sensor data to present to the user,
- the types of questions that should be asked to the user in response to the sensor data,
- the information to store in the knowledge base to help provide the information to evaluate the answers the system provides,
- the appropriate metrics to use in evaluating the answers (including the weights to put on each of the factors described in the previous paragraph),
- the details of the communication specifications between the system and the test harness, and
- the interfaces and the representation of the information to be passed between the various internal components of the framework.

However, for any of these components to be developed and tested, the overall framework must exist. Therefore, the development and implementation of the overall framework of the test harness, with initial black boxes for each of the individual components, is the first priority and thus is currently being developed.

Additional future work will focus on applying the test harness to other aspects of the autonomous system architecture (as discussed in Section 2). To be more specific, in this paper we only focused on testing the system's world model capabilities. However, we could expand the parts of the hierarchy being tested such that we allow the system to generate plans, and compare those plans to "optimal" plans as determined by the system's knowledge base which contains "perfect" world knowledge. We could also test the autonomous system's sensory processing components, by feeding in raw sensory data, and ask the

[2] Ideally, one would represent processing speed in independent unit, where the actual time could be based on multiplying the units by the appropriate processing speed factor.

autonomous system questions based on the processing of that data.

It would also be interesting to apply this test harness to other architectures besides RCS. Although, in theory, there is nothing RCS-specific about this architecture, it would be interesting to see how well the design holds up to other architectures for autonomous systems.

References

[1] Albus J., "4D/RCS: A Reference Model Architecture for Demo III, Version 0.1," NISTIR 5994, National Institute of Standards and Technology, Gaithersburg MD, 1998.

[2] Arkin R., *Behavior-based Robotics*, Cambridge, MA: The MIT Press, 1998.

[3] Brooks, R. "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14-23, 1988.

[4] Shoemaker C., and Bornstein J., "The Demo III UGV Program: A Testbed for Autonomous Navigation Research," in the Proceedings of the IEEE ISIC/CIRA/ISAS Joint Conference, 1998, pp. 644-651.

[5] Gazi V., Moore M., Passino K., Shackleford W., Proctor F., and Albus J., *The RCS Handbook: Tools for Real Time Control Systems Software Development*, New York: John Wiley and Sons, 2001.

[6] Albus J., and Meystel A., *Engineering of Mind: An Introduction to the Science of Intelligent Systems*, New York: John Wiley and Sons, 2001.

[7] Ganter B., and Wille R., *Formal Concepts Analysis: Mathematical Foundations*, Germany: Springer, 1999.

[8] Gruninger, M., and Fox, M.S., "The Role of Competency Questions in Enterprise Engineering" in the Proceedings of the IFIP WG5. 7 Workshop on Benchmarking - Theory and Practice, Trondheim, Norway. June 1994.

[9] Deborah L. McGuinness, Richard Fikes, James Rice, and Steve Wilder. "An Environment for Merging and Testing Large Ontologies." in the Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, USA 12-15 April 2000.

[10] A. Turing. Computing Machinery and Intelligence. *Mind,* 59, pp. 433-460. Reprinted in E. Feigenbaum and J. Feldman. (Eds.) *Computers and Thought*, McGraw-Hill, New York, 1963.

[11] D. K. Harmon (Ed.). The First Text Retrival Conference (TREC-1). NIST Spec. Pub. 500-207, National Institute of Standards and Technology, Gaithersburg MD, 1993.

[12] A. Jacoff, E. Messina, and J. Evans. A Standard Test Course for Urban Search and Rescue Robots. in Measuring the Performance of Intelligence of Systems: Proceedings of the 2000 PerMIS Workshop August 14-16 200. NIST Spec. Pub. 970, National Institute of Standards and Technology, Gaithersburg MD, 2001.

[13] William Mettrey. A Comparative Evaluation of Expert System Tools. *Computer* pp. 19-31. Feb. 1991.

A Native Intelligence Metric for Artificial Systems

John Albert Horst
100 Bureau Drive, Mail Stop 8230
The National Institute of Standards and Technology (NIST)
Gaithersburg, Maryland, USA 20899-8230
john.horst@nist.gov
voice: (301)975-3430

ABSTRACT

We define native intelligence as the specified complexity inherent in the information content of an artificial system. The artificial system is defined as a system that can be encoded in some general purpose language, expressed minimally as some finite length bit string, and decoded by a finite set of rules defined *a priori*. Using this definition of native intelligence, we employ a chance elimination argument in the literature to form a simple, but promising native intelligence metric. Several anticipated objections to this native intelligence metric are discussed.

1 INTRODUCTION

We define two perspectives on artificial system intelligence: (1) native intelligence, expressed in the specified complexity inherent in the information content of the system, and (2) performance intelligence, expressed in the successful (*i.e.*, goal-achieving) performance of the system in a complicated environment. In this context, complexity is simply Shannon information [5]. Specified complexity is Shannon information matched with meaningful patterns (*e.g.*, orthography, syntax, and semantics). Native intelligence aggregates things like potential intelligence, learning ability, system integration, richness and potential effectiveness of individual behaviors, and intellectual reasoning capabilities. These are fundamental, theoretical, and innate aspects of intelligence. The performance perspective on intelligence aggregates things like the efficiency of design and maintenance, real-time performance, and the ability to effect desired physical changes on the environment. These are external behavior characteristics that can be measured without knowing where the intelligence came from, how it is represented, or what algorithms are used internally to process the data and produce effective output.

These two perspectives have a parallel with the distinction between theoretical and experimental sciences. In this regard, the native and performance perspectives are complimentary and not competitive. They also ought to give the same results when applied to the same quantities in identical systems. Each one ought to inform and contribute insight to the other.

The essence of intelligence is completely non-material. This must be true, of course, if a native intelligence perspective is to have any validity. Intelligence can be embodied, but it is completely independent of embodiment. For example, a finite state machine (FSM) for a particular behavior can be stored on a compact disk, located in the mind of the designer, or spoken out loud to an audience. The FSM is completely independent of the medium of storage, type of representation, or mode of transmission. It exists as information. The quantitative measurement of intelligent systems is becoming and will become an increasingly important thing to be able to do. The discovery of DNA in living things as the living blueprint of life forms an existence proof that the essence of living things is non-material information. DNA is merely the medium; the information is the real intelligence[1].

1.1 The value of a native intelligence metric

A metric for quantifying the intelligence of a system *independent of its performance*, *i.e.*, a native intelligence metric (NIM), is needed. A NIM would bring substantial gains beyond that possible from performance intelligence metrics alone. A measurement of native intelligence can be made prior to the simulation or execution of the actual system, since all that is necessary to apply the metric is the representation of the system (as a string of bits), the rules for interpreting the string, and the meaning matching the system (the semantics). To measure success earlier in the

[1] Actually, the concept that all hereditary characteristics come from the DNA and *only* the DNA (called Crick's Central Dogma) has been shown to be false [1].

design phase is known to be critical to successful engineering design, particularly for large scale, complex systems. This is true even if the NIM is suboptimal, namely, even if some intelligence actually in the system is missed by the NIM. The ability to define and measure intelligence merely from its native intelligence offers promise to improve important quantities such as system time-to-market and quality. A NIM will also allow a more straightforward debug of the system, since the problem and solution will be more localized. For example, if we know that a few bits in the string cause the problem, we may be able to easily fix the problem by correcting just those few bits.

1.2 The relationship to living systems

Our definition of native intelligence of artificial systems is substantially unlike what is known as "IQ intelligence." IQ intelligence is a type of performance intelligence (intellectual mostly) in which the result is used to measure the potential of the human subject. Furthermore, a fundamental difference between living organisms and artificial systems is that the latter are completely malleable. This is not just a difference but a distinct advantage. Therefore, IQ intelligence cannot and should not constitute a model for the native intelligence of artificial systems.

In human children we often want to distinguish between the potential of a child to learn and how much they have actually learned. Being unable to do this is a source of incredible frustration. The problem is that we do not yet understand how to measure native intelligence of humans. I suspect that some day, perhaps soon, we will be able to make some level of a determination simply be interpreting the genetic information in the genome of each individual. This subject is the topic of a film entitled, Gattaca [2][2], in which the protagonist overcame his genetic "destiny" in spite of an oppressive determinism in the surrounding culture. With humans this is a loaded topic, but with machines the situation is much more amenable, since we can design the system for analysis! In fact, we want to design the system for analysis. We want the "intelligence meter" to detect the maximum amount of intelligence, and since we have full control on how we encode the description of the design of the system (just like we can always encode a computer program into a finite sequence of bits), we will choose to encode it in a standard format so that the intelligence is measurable.

With the NIM are we claiming that a system's performance is fully constrained by its initial determining information? Isn't this crass determinism? No, it is not. A system's performance is not fully constrained by the disembodied information describing the system, since performance depends on the nature of the environment, which will affect future performance. For example, if an adaptive filter adjusts parameters for a certain environment, subsequent exposure to another environment may cause instabilities in the filter. High native intelligence does not mean that the intelligence is realized. The NIM is not meant to exclude the system's potential response to some future contact with its environment. There is a multitude of ways that the same intelligent system can by realized. Finally, living systems are still a great mystery and are arguably not fully defined by algorithms (in the way that we are defining artificial systems) [3].

1.3 Specified complexity and computational theory

Measuring native intelligence is much like the problem scientists and mathematicians asked around the turn of the century. What are the natural classes of computing machines and can we measure the power of a machine? Are there limits of computation and if so what are they? Similarly our question becomes the following. How can we measure the intelligence of a system without ever seeing it perform or even simulating its operation?

In a manner consistent with definitions of information in standard computational theory texts, we propose that the minimal information describing a system entails its intelligence, *i.e.*, the intelligence is fully contained within the information describing the system [4]. Of course, by information we do not mean Shannon information, namely, that the information is merely complex [5]. Random noise is complex, but is not specified (*i.e.*, it has no "meaning") and therefore conveys no real (useful) information. Rather, what we mean by specified complexity is that the information is both complex and has discernible meaning, which is to say that it has recognizable patterns that indicate the power of the system to perform complex and useful tasks. This specified complexity can be represented and analyzed without actually realizing the system and seeing it perform. We argue that intelligence can be distilled down into pure information, even a finite amount of information. The description of a computational machine can always be reduced to a finite string of 0's and 1's. To represent a truly intelligent system, the system's information must be both complex (in the Shannon sense) and specified. In computation theory, information content of a machine is defined as the minimum representation of that machine. On the other hand, certain machines may look complex, but they may not do anything useful. Therefore, the system's description needs to be specified. It

---

2 Certain commercial companies and their equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply any judgement by the National Institute of Standards and Technology concerning the companies or their products, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose

needs to have meaning that, in some way, is grounded in reality or truth. And furthermore, we have some way of measuring the amount of this connection, *i.e.*, the greater the system connects with real meaning, the greater the intelligence. Here then is a method for measuring the native intelligence of systems: Measure the information content, using information theory and computation theory and measure the magnitude with which it can be matched to patterns representing truth-grounded semantics (something like a dictionary would be employed). For example, the DNA string is certainly complex, *i.e.*, there is no known simple representation for the entire string (no simple equation generates the string of nucleotides) and DNA is also certainly associated with known truth-grounded semantics, *i.e.*, it can be used to form proteins that are the machinery and computing engines of the most complex factory yet discovered, the living cell. Clearly, detecting and measuring the connection of a system's information with truth-grounded semantics is a substantial challenge.

1.4 The analogy to linear systems and the mixed success of intelligent systems

To generate an effective NIM, some mathematically sound concept of what fundamentally constitutes intelligence is needed. Why not use some of the existing IS models, such as neural nets, evolutionary programming, hierarchical models, behavior-based control, and fuzzy control? Because all these models share in common that they are primarily models for facilitating IS design and execution and do not in themselves constitute formal models of intelligence that will readily yield a NIM.

Why not use linear systems theory as a model for a NIM? The successes of traditional linear system theory can in many ways be considered the target we would aim for in the development of a useful NIM. In traditional linear system theory we have a most agreeable situation. The dynamic behavior of many systems can be described as differential equations of motion and dynamics and, coupled with state space structures and linear control theory, we can determine absolutely whether the system is controllable, stable, or observable without doing any simulation or building and testing of an actual system. However, it is well known that when we add non-linearity and time variance, for example, we can measure the stability of only a small subset of the total space of all non-linear systems. The mathematics also gets substantially more complicated and sophisticated. A large class of real-world systems (including discrete event systems) do not seem to submit easily to concise and simple mathematical descriptions. So this path (employing differential equation models) for determining the intelligence of systems has hit a fundamental barrier, it seems.

What about using formal methods as a basis for a NIM? The NIM offered in this paper may have much in common with formal methods and further study of this connection should be pursued. Formal methods have had a very limited impact, but this may not be due to any limitations in the theory. The limited impact seems to be due to the fact that many systems do not submit easily to the formal syntax of predicate calculus and that the learning curve is too high for most system designers to comprehend formal methods in order to successfully use them. Tools for formal methods also seem to be lacking or need substantial improvements [6].

The field of intelligent systems research has rightly been criticized by the traditional control systems community as being open to the proliferation of *ad hoc* design techniques, such as neural control, fuzzy control, hierarchical control, etc. These techniques are not grounded in physical and mathematical theories in the way that traditional control systems are grounded in the theory of dynamics and differential equations. There is certainly no commonly accepted equivalent to differential (or difference) equations for large-scale discrete-event systems (large-scale discrete-event systems may be considered to be the domain of IS). Traditional control systems can be measured in terms of their stability, controllability, and observability without realizing (or embodying) the system. These are metrics that allow us to measure the native intelligence of traditional control systems. For example, a system that goes unstable in a certain critical region would intuitively be less intelligent than one that has no such instability but in every other way performs equivalently. We have no such general metrics for measuring the native intelligence of large-scale discrete-event systems nor do we have the capability to compare the native intelligence of systems designed according to different methods. It is important to have quantitative measures for the relative performance of systems designed using neural control versus those designed by fuzzy control, etc.

Certainly *ad hoc* systems are helpful and even necessary at a time when there is no accepted theory with which to build non-*ad hoc* systems. However, without a theory, we often witness a proliferation of inflated claims of system performance that can not be met. This has several time produced a backlash from funding organizations who made financial commitments based on these groundless claims. This is all the more reason why a well grounded NIM would be helpful.

Furthermore, intelligent systems will have to be designed for analysis. But this is exactly what we do when we design linear, time-invariant control systems in a state space form. It's easier to analyze that way. Furthermore, vendors of systems will not want to hide the intelligence of their systems, but will want others to know for sure that their system is truly intelligent. The advantage of having a

NIM should be obvious. That way we can measure intelligence prior to any commitment to simulation in software or testing in hardware. Computation theory can also help with measuring the degree to which the information (describing the system) matches with truth-grounded semantics. For example, a system that can recognize or distinguish between a broader set of input "languages" is more powerful than one that cannot. For example, a finite state automata (FSA) can be designed to recognize strings of the form 0^*1^*, but no FSA can be built that recognizes strings of the form 0^n1^n, whereas a push-down automata (PDA) can be built to recognize strings of either type. Therefore, the PDA is more powerful (intelligent).

Measuring intelligence is similar to what a cryptanalyst does normally except that when measuring intelligence we don't anticipate the element of planned deceit. The cryptanalyst is looking for signs that the bits have identifiable patterns underlying the randomness. If there is no randomness, there are no patterns, and no intelligent message underneath. With randomness, the bits could actually be truly random and therefore be nonsense. However, it may just appear to be random and the patterns are not obvious to the cryptanalyst. Ostensibly, an IS would be designed not to fool the IS metrics analyst to think the system is nonsense, but rather the IS would be designed so that its intelligence would be easily perceived by all.

1.5 The nature of a NIM

Perhaps the solution is not in an analogy to linear systems theory, as has been the hope, but rather in information theory, probability, and complexity theory. This is the research question we have asked ourselves and hope to provide some direction towards an answer. We argue that the essence of intelligence in living things is information (versus physics or chemistry). If this is so, we need to discover appropriate tools to correctly analyze that information to glean from it the level of intelligence in the system.

We claim that chance, regularity, and intelligence are mutually exclusive. Regularity is indicated by signals of low complexity. Therefore simple Shannon information filters can eliminate regular systems from consideration as intelligent systems (recall our definition of native intelligence as complex and specified information, in which the level of complexity and specification is proportional to the level of intelligence). After first eliminating regularity, if we can measure the probability that a system arose from chance processes alone, then the intelligence in that system must in some simple way be related to that probability. In a manner similar to Shannon information theory, we propose that, if there is a measure of the probability that doesn't just measure the mere complexity of the information (Shannon), but measures the *specified* complexity of the information, then the expression $-\log_2 p$, where p is the probability that the supposed specified complexity (of the system) arose purely through a logical chance hypothesis, is a theoretical measure of the system's native intelligence. Such a measure exists [7]. We will describe and examine this measure and suggest how it applies as a NIM.

2 THE CHANCE ELIMINATION ARGUMENT

Our goal is to find a quantitative intelligence metric for an artificial system that can be applied to the minimal informational description of the system. Another way of looking at this problem is to ask ourselves, what is the probability that this system either arose merely by chance (or merely by regular processes)? Regularity is relatively easy to eliminate from consideration since the information can in this case be generated from a relatively small expression (or a small number of "lines of code"). It may contain many bits, but not have much information. An example of regularity is an infinitely alternating string of four 0's and four 1's. The lines of code are simple: write down four 0's, write down four 1's, repeat steps one and two. Clearly there is little intelligence in this system. So, having eliminated regularity, if the probability that chance did the work is high, the intelligence in the system will be low; if the chance probability is low, the intelligence will be high. Why is this so? There are only three options for the source of artificial system information: chance, regularity, or intelligence. It has to be one of the three. So if chance and regularity have sufficiently low probability, then the source must be intelligence. Say that the system needing analysis is encoded in what appears to be a random sequence. If we happen to know that it is the description of a complex space shuttle control system, we can readily eliminate chance and declare it to be originating from intelligence. Note that false negatives are possible (thinking it is a chance process when it is an intelligent process), but false positives are avoidable.

A theory for chance elimination has been developed by Dembski in [7] and is called the Generic Chance Elimination Argument (GCEA). This theory forms the basis for the NIM development in this paper. Therefore, we will begin with a summary of the aspects of GCEA relevant to IS metrics. In the next section, we will investigate how the theory might be (simply) applied to IS metrics.

We start with a subject, S, that observes an event, E. By analyzing the circumstances surrounding E, S defines the chance hypothesis, H, gotten from a reasonable chance process that might have been responsible for E. S discovers (doesn't matter how) a pattern, D, that matches the event,

E^3. With D* as the event associated with the pattern, D, S calculates the conditional probability, P(D*|H), of the event, D*, assuming the chance hypothesis, H, is true. S tests whether P(D*|H) is less than a universal probability bound, δ. Dembski determined that $\delta = 10^{-150}$ from fundamental physical limits (space, particles, and time) within the known universe[4]. Through knowledge of some side information, I, S computes the conditioned complexity $\varphi(D \mid I)$ of formulating the pattern, D, given the side information, I. "Complexity" in this context is defined within the domain of "complexity theory," the most common example of which is computational complexity. Complexity theory is a dual with probability theory in the sense that with probabilities we are dealing with "events" conditioned by "background information" and with complexity theory we are dealing with "problems" conditioned by "resources." So formulating D is the "problem" and I are the "resources" for solving the problem. A problem conditioned by resources places $\varphi(D \mid I)$ in the domain of complexity theory. S computes a tractability bound, λ, which is used to calibrate the complexity values, since complexity, unlike probability, is essentially uncalibrated. λ is the upper bound of complexity, below which the side information, I, is sufficient to form the pattern, D. Finally, if P(D*|H) < δ and $\varphi(D \mid I) < \lambda$, S can infer that E did not occur according to H. This is a substantially abbreviated form of Dembski's argument, but is sufficient for our needs. We will now give an example to help clarify the GCEA.

Say we are S and we stumble upon Stonehenge. We don't wonder whether humans carried the stones (some weighing over 5×10^4 kg) to the site. However, we do wonder whether the particular arrangement of the stones is meant to align with certain seasonal, celestial events such as the positions of sun and moon at summer solstice and during eclipses. S declares that E is the particular arrangement of stones S encounters that may be exhibiting alignment with celestial bodies at certain seasonal times. S determines that the designer of Stonehenge had full view of the sky enough days in the year to be able to note seasonal occurrences. S further determines that at summer solstice, for example, the rising sun precisely aligns over the Heel Stone shining rays directly into the center of the monument in the inner horseshoe arrangement of stones. S notes that there are several of these curious alignments of lunar and solar events (including eclipses) with various stones. This alignment of events with stones constitutes D, our pattern. D* then constitutes the particular pattern of stones consistent with the alignments. E would then, at the least, be delimited by D, but for simplicity's sake, say D* = E. Now we calculate the probability, P(D*|H), that D* could happen by chance alone, even though D* was gotten by assuming the matching of the stones with particular celestial events. The various celestial events and our prehistoric Stonehenge designer's awareness of these events constitute the side information, I. We calculate the tractability bound, λ, below which the particularly alignment of stones is possible given I. We then calculate $\varphi(D \mid I)$, the actual complexity of forming the pattern given I. If P(D*|H) < δ and $\varphi(D \mid I) < \lambda$, S can infer that the particular arrangement of aligned stones, E, did not occur according to the chance hypothesis, H.

All these definitions and formulations are completely consistent with probability theory, complexity theory, and statistics. Note that the measurement is heavily dependent on the correctness of I (*i.e.*, that I is statistically "sufficient"). For example, we must have sufficient knowledge whether the prehistoric Englishman could perceive those celestial objects and that the alignments have changed but slightly over these thousands of years. On the other hand, say we now know that the atmosphere of England was even more persistently overcast thousands of years ago than it is today. We might then conclude that $\varphi(D \mid I) > \lambda$, *i.e.*, our prehistoric Englishman caught sight of the celestial bodies so infrequently that the complexity of forming the match with the alignment pattern would be virtually impossible. Now a possible criticism here is that our knowledge of the prehistoric Englishman's weather conditions may be imprecise. In this case we may only be able to determine a range for $\varphi(D \mid I)$. All that would be necessary then is to ensure that the upper end of the range is less than λ, the complexity tractability upper bound.

Furthermore, even if the complexity happens to be tractable, *i.e.*, $\varphi(D \mid I) < \lambda$, let us say that only one stone is involved in the alignment pattern. This would mean that P(D*|H) would be quite high, since the random spatial arrangements of only one stone are relatively few compared to the random spatial arrangements of N stones with N >> 1. Therefore, it is conceivable that P(D*|H) > δ, particularly since δ is such a conservatively small number, even though $\varphi(D \mid I) < \lambda$.

3 Chance Elimination Argument transformed into a NIM

3 Actually if D* is the event associated with the pattern, D, and the occurrence of E implies that D* also occurred, we say that D *delimits* E rather than D matches E.

4 There are less than 10^{80} elementary particles in the known physical universe, no more than 10^{45} physical state transitions per second, and (assuming "big bang" cosmology) no more than 10^{25} seconds of time will ever be available in the known physical universe. So $10^{80} \times 10^{45} \times 10^{45} = 10^{150}$ is a liberal measure for the maximum number of possible state transitions of all the particles in the known universe throughout the total lifetime of the universe.

Our goal is to apply the GCEA to the problem of developing a NIM. The GCEA is intended simply to *eliminate* the chance hypothesis from consideration, whereas we wish to discover a metric. However, in the process of eliminating chance, the GCEA has to compute a measurable quantity, namely, the probability of the pattern (transformed into event space) conditioned by the chance hypothesis, P(D*|H). The GCEA is also intended merely to compare P(D*|H) against a threshold δ, in order to (possibly) eliminate the chance hypothesis and infer intelligent agency. However, the artificial systems we normally wish to measure are those that we know intelligent agents have designed *a priori*, so no threshold comparison is needed. Probabilities are already calibrated, having a target range in the interval, [0,1]. The GCEA computes a complexity measure, $\varphi(D \mid I)$, comparing it with λ, in order to ensure that the pattern is tractably constructible from the side information. Even though we know that our system is designed, it is certainly possible that it cannot be executed. Therefore, the role of the complexity measure should remain the same for the IS measure. Finally, the GCEA computes a *probability* and we are interested in a complexity measure. Since regularity can be easily eliminated from consideration, the only other options are chance or intelligent agency. Therefore, to the degree our candidate IS is not attributable to chance, the IS displays intelligence. Furthermore, the information theoretic method for converting a probability into an information measure is to take the negative of the logarithm base 2 of the probability. Since we've defined information as the specified complexity of the minimal representation of the IS, $-\log_2 P(D^* \mid H)$ is the NIM we seek. So a larger NIM means higher system intelligence and *vice versa*.

Therefore, the NIM consists the following steps: 1) identify E as the design of the IS of interest, 2) gather all the relevant side information, I, in order to identify the pattern, D, 3) identify the event D* that delimits E, 4) define the chance hypothesis, H, for the event D*, 5) compute the probability, P(D*|H), that D* might occur according to H, 6) compute the tractability bound, λ, by quantifying the complexity of forming the pattern with the minimal amount of resources, 7) compute the complexity of generating the pattern, D, from I, 8) test if $\varphi(D \mid I) < \lambda$, if true, then continue to next step, or if false, stop (it has no intelligence if it is impossible to construct and/or execute), 9) compute $-\log_2 P(D^* \mid H)$.

How do we identify E? This is simply the particular system representation we wish to analyze. However, the system design needs to be in an analyzable format. This begins with a need for some unambiguous syntax that can be represented in something like the extended Backus-Naur form (EBNF) [8]. How do we identify I and D? The relevant semantics will be captured in the side information, I, and the gathering and identification of side information is expected to be a challenging task. The challenge should lie mostly in the difficulty in assembling all the applicable semantic information. In fact, we can easily underestimate the intelligence in the system if we miss or overlook key information. Going back to the Stonehenge example, if we mistakenly conclude that the prehistoric man is inobservant and therefore is completely ignorant of eclipse events, but the eclipse alignments actually match with several alignment of stones, we will conclude that the stones display less intelligence in their particular arrangement than they actually do. This is a false negative, which will be very common. However, because complex specified information is typically highly differentiated to other complex specified information (*e.g.*, Schubert's music is very different from J.S. Bach's), false positives will be highly unlikely. The use of formal and standard system specifications [9] should also be used in order to define what is meant by the syntax. Therefore, intelligence in the system will sometimes go unnoticed. Different than the Stonehenge example, system designers will want to ensure that the systems they design achieve the maximum (intelligence) value under the NIM. This is called design for analysis and is done all the time now, though more for performance metrics, since they are most common. However, the current practice of formal testing can be generally considered in the same category as the NIM [10].

4 Objections to the NIM

A possible objection to the NIM is that a system such as a neural net or an evolutionary algorithm starts out without a lot of intelligence, but as it learns, it grows in intelligence. Recall that we specifically do not define intelligence as the ability to realize certain tasks, but rather as the specified complexity inherent in the system at design time, prior to execution. The mere *ability* to learn should be considered intelligent, independent of, or prior to, having learned anything. The simple acquisition of knowledge, even behavioristic knowledge, should not be counted as an increase in intelligence. For example, a person with advanced Multiple Sclerosis may be intelligent even though he or she may neither see, speak, nor move.

Another objection to the NIM is that certain intelligent systems, particularly learning and optimization approaches such as neural nets or evolutionary algorithms actually effect an increase in specified complexity as the system executes, because it solves complex optimization problems without front-loaded intelligence. However, it can be shown that through the designer's choice of the particular fitness functions, system structure, and initial populations, specified complexity is entered surreptitiously [11]. The No Free Lunch theorems are proof of this, since, for any particular optimization algorithm, if the fitness functions are not constrained within a certain domain, but are allowed to freely range throughout the entire domain of possible fitness functions, the optimization algorithm will not perform on the average any better than blind search [12]. So specified complexity can only be added by intelligent agents [13].

Another response to this objection is largely experiential. The author has now some fifteen years experience in the practical design of large-scale control systems for mining vehicles, inspection machines, and on-road autonomous vehicles. The consistently strong impression is how much sophisticated, up-front design intelligence (from the intelligent agent) is needed to generate a successful system. Furthermore, the maintenance and improvement of the system also requires enormous input of human intelligence, very far from the easily and powerfully evolving systems that are promised. Perhaps though, we have been using the wrong design paradigm (we have been using a hierarchical paradigm [14]). Because of the No Free Lunch Theorems, we suspect that switching to another design paradigm will make little difference in this. Furthermore, as a working intelligent system designer, we have, of course, had substantial interaction with other designers using other paradigms. Based on this informal testimony, we have no reason to believe that the other paradigms require *substantially* less input of external intelligent agency, both at design time and after.

Yet another objection is that performance intelligence metrics are easier to obtain and furthermore are all that is needed. Besides the NIM is too hard to compute and obtain, *i.e.*, too much work is required to make it worthwhile. This could be argued against linear system theory, which certainly requires a substantial mathematical and physical sophistication, and very few would argue that it has not been of substantial value to control system design. How is linear system theory like the NIM we propose? Linear system theory supplies an analytical theory and allows for metrics for measuring native capability, such as stability, controllability, and observability. Similarly, NIM is based on probability, complexity, and statistical theory and provides its own system capability (or intelligence) metric. As to the objection that the NIM is too hard to compute and obtain, just in the same way that the effort of converting homogeneous differential equations into state space form allows us to apply metrics, converting our IS into an appropriate format, should also allow ease of analysis.

One might argue that elegant and simple designs performing some function should be considered more intelligent than more complex designs that perform exactly the same function. In this case we can consider as side information some metric like "design efficiency" that will increase $-\log_2 P(D^*|H)$ for the more elegant design.

Will the NIM perform well given a complex but useful system as well as an equally complex system but one that simply thrashes, performing no useful task? The key here is that $P(D^*|H)$ will be higher in the latter system, since the thrashing system is not specified even though it is complex. The transformed pattern, D^*, will not be supported by side information, I, and so the chance hypothesis, H, will have increased support. Additionally, how will the NIM perform given two other systems, one buggy and one bug-free? Again the system with the bugs will fail to support the side information (that is, the side information affected by the bugs) that indicates increased intelligent agency, giving the chance hypothesis increased support.

Finally, it will be argued that chance, regularity, and intelligence do not cover the entire space of possibilities and particularly that intelligence is not the only conclusion when there is specified complexity. Both Kaufmann [15] and Davies [16] argue that there must be an (as yet) undiscovered regularity that can generate specified complexity. Since this is not yet discovered and since there is little concrete evidence for its existence, it certainly is not unreasonable to maintain our claim.

5 CONCLUSIONS AND FUTURE DIRECTION

We make no claim that the determination of the actual value of the NIM will be easily gotten, particularly as the systems under analysis become more complex. The precise determination of side information, the matching pattern, and the event probability under the chance hypothesis will be challenging to quantify with reasonable hope of accuracy. Clearly, many examples, starting with known and relatively simple systems (such as linear systems), should be attempted in order to exercise this metric.

We may also discover that such a metric (or any intelligence metric, for that matter) cannot measure certain broad classes of systems. This would be disappointing, but should not stop us from pursuing a metric or even surprise us. The twentieth century has been known for the discovery of a wide variety of limitations. Special relativity posited a limiting speed for matter. Heisenberg discovered a fundamental uncertainty in measurement capability (infinite measurement accuracy in position and momentum simultaneously of a particle is impossible). Chaos theory realized that the trajectories of certain deterministic systems cannot be accurately predicted without infinite precision in the initial conditions. A broad class of problems is either not computable (like the general tiling problem) or unsolvable given our computing resources (like NP-hard problems). All these are extremely helpful discoveries, even if somewhat disappointing. At a minimum, they lessen inflated claims of performance by quantifying performance limitations. Perhaps some similar fundamental limitation for any intelligence metric is also inherent in some broad class of artificial systems. To find such a limitation and to parameterize it would be a worthwhile discovery.

REFERENCES

[1] Barry Commoner, "Unraveling the DNA myth: The spurious foundation of genetic engineering," Harpers, February, 2002.

[2] Columbia Tristar Pictures

[3] Roger Penrose, "The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics," Oxford University Press, 1989.

[4] Michael Sipser, Introduction to the Theory of Computation, PWS Publishing, 1997.

[5] Claude E. Shannon, "A mathematical theory of communication," Bell System Technical Journal, Vol. 27, July and October, 1948.

[6] John C. Knight, Colleen L. DeJong, Matthew S. Gibble, and Luis G. Nakano, "Why are formal methods not used more widely?" Fourth NASA Formal Methods Workshop, vol. 27, September, 1997.

[7] William A. Dembski, The Design Inference: Eliminating Chance Through Small Probabilities, Cambridge University Press, 1998.

[8] ISO/IEC 14977:1996(E), Extended BNF, ISO, Geneva, Switzerland, 1996.

[9] John A. Horst, Elena Messina, Thomas Kramer, and Hui-Min Huang, "Precise definition of software component specifications," Ghent, Belgium, 1997.

[10] Paul E. Black, Kelly M. Hall, Michael D. Jones, Trent N. Larson, and Phillip J. Windley, "A brief introduction to formal methods," IEEE 1996 Custom Integrated Circuits Conference, 1997.

[11] William A. Dembski, No Free Lunch: Why Specified Complexity Cannot Be Purchased without Intelligence, Rowman & Littlefield, 2002.

[12] David H. Wolpert and William G. MacReady, "No free lunch theorems for optimization," IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, April, 1997.

[13] Douglas Robertson, "Algorithmic information theory, free will, and the Turing test," Complexity, vol. 4, no. 3, 1999.

[14] J. S. Albus and A. M. Meystel, Engineering of Mind: An Introduction to the Science of Intelligent Systems, John Wiley and Sons, 2001.

[15] Stuart Kaufmann, At Home In The Universe: The Search for Laws of Self-Organization and Complexity, New York: Oxford University Press, 1995.

[16] Paul Davies, The Fifth Miracle, New York: Simon and Schuster, 1999.

Dimensions of Intelligent Systems

Dr. Gary Berg-Cross
Knowledge Strategies Division, SLAG, Inc.
Potomac, Maryland 20854, USA

1. Abstract

As intelligent systems have become more fully functional and commonly available, questions about their capabilities and relationship with humans have increased. This paper builds on the IS requirements ideas of Messina et al [2001] to explore middle ground between anthropomorphic approaches like the Turing test that rely on similarity to human behavior in an "imitation games" and the narrowness of tests of chess mastery. I contrast a system like Deep Blue which has a very fixed environment in which it performs to more complex types such as Associate technology. Deep Blue, I argue, is an example of system whose performance is expert, but whose competence is fragile and it may not satisfy extended definitions of competence and performance intelligence that we measure in dynamic environments. A clinical protocol system is used to explore the basic functional capabilities and knowledge. Beyond symbol processing and the knowledge level are grounded reactive intelligent within more of an environmental/systems perspective. I build on grounded systems to discuss the use of goals using, learning-based systems and & multi-modal logics that characterize "realistic" intelligent systems. It is argued that such characterizations will evolve as IS design matures into grounded intelligence and situated, rational agent systems. In the future belief models and measures of rational coherence might be used, as basic approaches to facilitate intelligent system performance in dynamic environments.

Keywords: IS, Intelligent Systems, Turing Test, Cognitive Model, situated cognition, BDI, Deep Blue, constructionism

1: Introduction

Investigation of artificial intelligence system capabilities now has a long history with notable discussion stemming from the original Turing test with many modern elaborations. Motivating contests exist for passing a test such as Turing's (Loebner Competition) as well as prizes for tasks in chess and various robot competitions (RoboCup, Office Navigation, Trash Pickup etc.). However, something like a Turing Test, imitation of human conversation, seems too difficult if we take it to logical conclusions, while success in something like chess as demonstrated by the triumph of Deep Blue seems too narrow an achievement to feel that we have made general progress on truly intelligent systems. Since the intersecting concepts of performance, intelligence and systems are complex, success with Intelligent Systems may be aided by focus on tasks of intermediate challenges. I will discuss several examples from the medical realm to illustrate the challenge and the state of affairs. From the example of patient safety performance I build a framework of intelligence perspectives or dimensions to organize the discussion. It has some clear idea of successful performance. For diagnosis we can ask if it is correct given ratings by an expert panel. Several IS diagnosis systems have been implemented and indeed, by expert ratings, they perform better than a typical physician, yet they have little penetration in the healthcare industry. Why? One reason is that the measure of correctness is isolated and doesn't look at the total picture including cost and maintenance issues. The system's knowledge can be difficult to maintain and further systems have difficulty fitting into the working environment, an issue I discuss more under the topic of Associate Systems.

Still another factor concerns the issue of patient safety and system error. There are 4 categories by which human performance is judged in relation to patient safety [Marx, 2001]. These are:

1. Human/system error
2. Negligent conduct
3. Reckless conduct
4. Knowing violations

We can see that judging human skill quickly gets beyond mere performance when assigning these categories[1]. Human/system error is a judgement that the system's performance was "inadvertent" and other than intended. We make such errors every day with minimal consequences and so might systems. The 2nd category, negligence, is a more culpable behavior and in healthcare is generally assigned when an individual has been harmed by a failure to exercise skill, care and learning expected of a provider [Marx, 2001]. Thus, we quickly leave a purely behavioral domain and enter one with concepts like "learning' skill and intentions. There are several distinct architectural levels that can be distinguished meaningful beyond the "what" of behavior that include why (a knowledge and cognitive level), how (functional /symbol) level and what descriptions. This leads to higher dimensions for judging behavior outlined in the four dimensions or views as shown in Table 1. At the

[1] For current implementations we could agree that ISes aren't going to make category 4 safety errors until we have systems whose intentions are explicit!

foundational level of behavior we can discuss the most obvious aspects of performance. When we talk about a system at this level the intent is not to go beyond the dimension of its behavior. But already there are many issues here, such as emerged from earlier attempts at "Behavioral Psychology". Our patient safety example illustrates the problems in principle. A second dimension, shown in Table 1 is a functional approach that is typically couched in a stimulus, information processing, and response model.

Dimension/Perspective	Characterization
1. Behavioral/What	Performance assumes not other system/dimensional knowledge. Such behavioral descriptions list the observable behavior exhibited by a system when it is being applied or executed. This model of system behavior (i.e., a series of episodes of the system's activities) relies on observation.
2. Symbolic/Functional Architecture (How)	Traditional information processing of symbols. The functional model describes an (implemented) representational and computational commitments/architectural primitives.
3. Environmentally Reactive (External why)	Intelligent behavior is a coherent response to environmental challenge. To do this it may functionally involve goals and world models. The knowledge level description describes a system in terms of the knowledge of the world and some principles that are applied when using that knowledge.
4. Goal-oriented and Intentional (Internal why)	Agent-orientation to rationalize intelligence at a belief/goal/ intention level. At this level we have refined principles of rationality.

Table 1 Perspectives of Intelligent Systems (What, how and why)

Prior PERMIS conferences in 2000 and 2001 provide a broad discussion on the testing of Intelligent System (IS) based both on behavioral performance, including efficiency and effectiveness measures drawing on the expectations of designers, as well as functional capabilities including robustness and learning capabilities etc. One way of pursing the question has been to take performance measures of non-intelligent systems and to attempt to add measures for intelligence [Messina et al 2001]. The simplest way to frame this has been to discuss the main elements found in IS. Messina et al [2001] propose several elements that make up a functionally intelligent architecture including:

- behavior generation to deal with incomplete commands (e.g. interpret commands, supplement instructions),
- synthesize alternative behaviors and adjust plans;
- adjust sensory processing to deal with the unexpected and unknown; and
- represent the world using an updatable, long-term stores of knowledge, including commonsense notions.

Such a listing fits the now classical Input, Process, Store and Output information processing type of model of intelligence such as shown in Figure 1. In such robotic models, perception and behavior are treated as separate, front end functions and "cognition" which, goes on in the processor and memory functions controls the perceptual and effector functions. This is a popular concept of intelligence as basically cognition [Newell, 1982]- the capacity to construct and manipulate symbolic representations, i.e. "approximate models" that are mapped to the environment and determine "appropriate" action. This is also called a knowledge-oriented view, since a system's knowledge is a way to describe behavior (e.g. synthesizing, adjust plans etc.). Chong and Berg-Cross [1990] provide an example of such work to understand the types of errors that ISes might make. Although models differ widely in terms of how sophisticated their concepts of knowledge, cognitive process and learning are Messina et al [2001] provide a useful base list several of functional requirements for testing cognitive systems e.g.. tests to measure the ability to fuse data from multiple sensors, including the resolution of conflicts. One of the goals of this paper is to follow up this approach by applying this criteria and additional characteristics raised in higher levels (3 and 4) discussed later or one advanced systems to illustrate assessment. A medical protocol planning system is used later in the paper to illustrate this.

There are many alternative ways of distinguishing the third distinction of intelligence. I call this Interactionist following a view of intelligence that see it, like knowledge, as open to interpretation and always relative to others things that provide an environment [Clancy 1989]. Steele [1995], for example, follows this view and sees self sufficiency in such interactions, which is called agenthood, as the basis for intellect. In this view we judge behavior as intelligent to the extent that it sustains an agent in an environment. Mail delivery robots are intelligent if they can successfully interact with and navigate a mail environment, especially if it is complex, uncertain and changes often. The key measurement is not a specific behavior but is described in terms of the quality of a result relative to this environment and what it presupposes. That result can be categorized as "success" in terms of a context. This is the same level of analysis as provided in the patient safety example previous discussed. Hence calling a system intelligent or its behavior intelligent, or in error, is based on an external, human/observe judgement as shown in Figure 1 [Van de

Velde 1995] rather than being a structural or even functional part of a "system". This shift in view includes the notion that knowledge is situational and cannot be viewed as self contained. Instead, it is inherently coordinated with an environment as situated context. If we buy this interactionist view of intelligence we quickly see the connection to a Turing Test (TT) which embeds system intelligence in a human context [(Saygin et al. 2000]. The Turing test is behavioral and interactionist but has a naive anthropomorphism implicit in a human "imitation" games. "If one were offered a machine purported to be intelligent, what would be an appropriate method of evaluating this claim? The most obvious approach might be to give the machine an IQ test . However, good performance on tasks seen in IQ tests would not be completely satisfactory because the machine would have to be specially prepared for any specific task that it was asked to perform. The task could not be described to the machine in a normal conversation (verbal or written) if the specific nature of the task was not already programmed into the machine. Such considerations led many people to believe that the ability to communicate freely using some form of natural language is both an essential attribute of an intelligent entity and a confirming test of underlying competence.

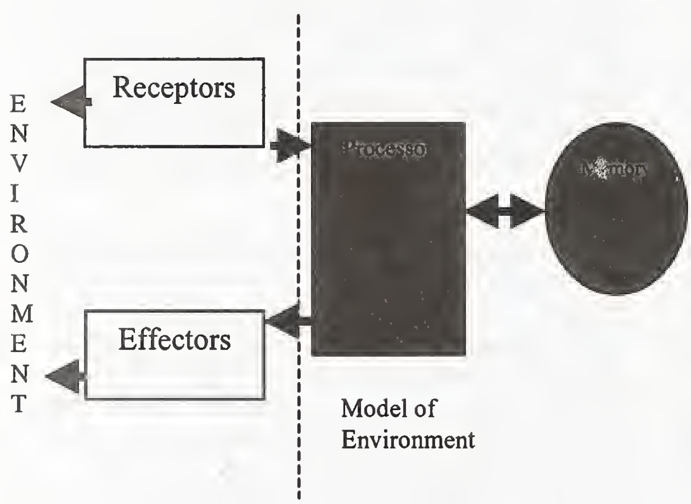


Figure 1 Model of Information/Symbol Processing

But philosophers like Searle with his Chinese Room argument challenge the Turing Test and its natural language exchange as a basis for assigning intelligence. This group maintains that the judgement of cognitive phenomena cannot be solely on the basis of observed input-output behavior. It is worth pointing out in passing that many strong AI critics like the Interactionist view, especially when it touches on the relative merits of symbolic learning and connectionist learning for implementing intelligence. Many find that Stevan Harnad's [1990] hybrid model to grounding symbols in the analog world with neural nets is a useful approach. The issue of the behavioral problems of the naive Turing test are taken up later in the paper in the context of Deep Blue's intelligent behavior.

Following Dennett's [1987] philosophical formulation, the fourth level explicitly considers intentions and belief, such as our patient safety . In philosophy the so called intentional stance position serves as a convenient, abstract way of talking about intelligent systems, allowing us to predict and explain their behavior without having to understand or describe how the cognitive mechanism actually works. Cognitive theorists and modelers have elaborated this in terms of cognitive structures that are capable of performing cognitive processes which in turn use those structures. Developers have to make many architectural decisions to actually implement such a philosophy. In the main this is the view a useful or "realistic" agent whether it is to be realized as "intelligent" software or as an autonomous robot. Such practical agents range from information gathering and trading agents to autonomous vehicles and may include real time physical capabilities, good for dangerous situations beyond the human central nervous system capacity.

The remainder of the paper is as follows. Section 2 walks through an example of an intelligent system applied to clinical guidelines to illustrate both the behavioral and functional criterion of Messina et al [2001] as well as the use of plans and goal reasoning in a system. Section 3 discusses the intelligence of Deep Blue using an interactionist robotics and unified cognitive architecture perspective. Section 4 returns to the Turing test and expands the concept to bridge to more useful concepts of judgement of intelligence and in particular the models that employ beliefs and intentions. Section 5 summarizes major findings, proposed initiatives arising from this view, dusts off the old concept of Associate Systems and proposes motivational competitions to enliven the field.

2. Gauging the Performance of Intelligent Clinical Protocol System

One of the most prolific areas of AI research has been in the medical realm which provides more representative measures of intelligence than chess performance and where there already exist regular monitoring efforts to judge success. A survey of the entire field is well beyond the scope of this paper, and even sub-areas, such as diagnosis are too diverse to cover easily. Instead I look at one well researched area - the support provided by intelligent system for clinical protocols/guidelines and a system that has been designed to aid people in developing and using clinical protocols also call guidelines. Clinical protocols have been developed over the last 12 years provide a quality standard of care for such things as diagnostic and therapeutic procedures, typically based on the consensus of experts. Protocols are increasingly in widespread use and The American medical Association's Directory of Practice Parameters listed over 1,500 several years ago. However, there is little sound data to judge effectiveness across medical practice and intelligent processing has been researched to automate, support and improve guideline-oriented medical care [Musen et al 1996]. One reason for selecting a clinical protocol system is that task analysis of the field has been conducted [Sharhar et al 1998]. along with system development such as by the Stanford Medical Informatics work with EON [Musen et al 1996]. That work includes support of tasks for:

- determining the applicability of a guideline for a given patient,
- generating recommendations for therapeutic interventions and lab tests via a protocol
- tailoring the recommendation to the context of the current patient situation and stage of protocol execution,
- monitoring the application of the protocol guideline,
- assessing the effectiveness of the guideline.

I use the Messina et al [2001] list of requirements for testing Intelligent systems along with the performance evaluation properties (I would call functional capabilities) for ISes in non-numerical domains list and Musen et al [1996] to illustrate several of EON's interesting features. EON but is built from general, purpose software components and has been applied to protocol-based care in domains as diverse as oncology, hypertension, AIDS and diabetes.

Requirement 1 & 13: "interpret high level, abstract, and vague commands and convert them into a series of actionable plans" and " to understand generic concepts about the world that are relevant to its functioning and ability to apply them to specific situations".

EON's main task is a general one, generate an acceptable plan given the current clinical situation and relevant guidelines. It must determine a patient's eligibility and refines abstract plans to fit the situation. A relevant property is its ability to deal with general and abstract Information. EON's designers recognized that it needs to be a general problems solver like a physician . Thus it deals with both detailed, patient data in the medical record and database and abstract protocol specifications. It infers higher-level, interval-based concepts using time-stamped, patient data. Conceptual abstractions are a major feature of the EON approach and the PROTÉGÉ II system is used to build the EON KBs emphasizing the use of conceptual abstractions to define problem-solving behaviors independently from the programming logic.

Also relevant is the ability to "deduce particular cases from general ones". EON contains time abstraction, general medical ontology (e.g. concepts and relations between prescription, drug regime, medication, clinical trial etc.) as well as disease and patient specific knowledge/information. It uses an episodic skeletal plan refinement method on very general time concepts to instantiate a patient guideline plan. An illustration of this is shown in Figure 2. This deduces particular plans from abstract information in combination with very specific information.

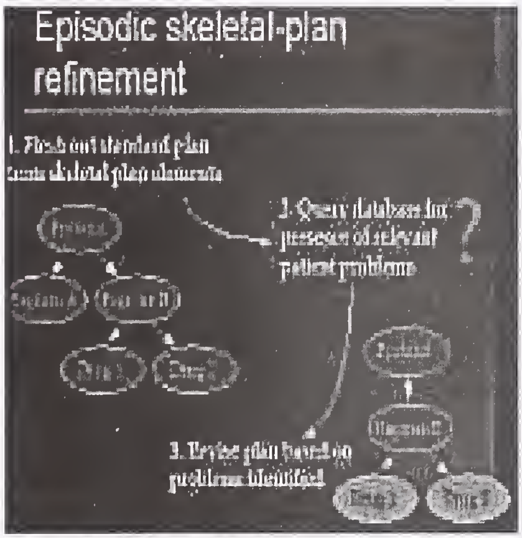


Figure 2 EON Skeletal Plan and Refinement Process

Requirement 2, 3 , 15 & 14: "to autonomously make decisions as it is carrying out its plans" and to "re-plan while executing its plans and adapt to changes in the situation" and " work with incomplete and imperfect knowledge by extrapolating, interpolating, or other means " and " deal with and model symbolic and situational concepts as well as geometry and attributes".

While EON is not an independent agent it's processing includes setting sub-goals as part of its plan refinement. An observer would see it going far beyond the original specification in several steps:

1. Identify and propose a starting standard, abstract hierarchical plan

2. Instantiate the plan based on situation and decomposition and to allow execution (time constraints etc.)
3. Identify problems that might make application of this plan (practical drug admin challenges, side effects, etc.)

A property of the system is the ability to reschedule and replan and adjust the plan to updated situations [Messina et al 2001]. It also might be said to "recognize the unexpected" in that the guidelines project out a path and it will replan if deviations occur such as reduce AZT if anemia develops or side effects develop. Similarly it deals with incomplete information routinely. It typically does not have the entire attribute value set to begin with and generates queries of relational DBs that are processed into patient history. EON deals with situations of time, but distance and geometry, such as robotic path concerns are not part of its knowledge base (KB).

Requirement 4-9: to "register sensed information with its location in the world and with a priori data " and " fuse data from multiple sensors, including resolution of conflicts " and " to handle imperfect data from sensors, sensor failure or sensor inadequacy for certain circumstances" and " to direct its sensors and processing algorithms at finding and identifying specific items or items within a particular class" and to "focus resources where appropriate" and " to handle a wide variation in surroundings or objects with which it interacts" .

EON does none of this. It is not robotic. An IS is often robotic in having sensor and/or effectors but many are decision supports. Adding a robotic element to an IS is discussed later in the context of interactions and a Total Turing Test.

Requirement 10-12: to "deal with a dynamic environment" and "map the environment so that it can perform its job" and " update its models of the world, both for short-term and potentially long-term".

As noted EON deals with changes in the patient situation as well as changes to guidelines and phase of care. However, it's function does not result in model-mapping of the situation as might be implied here. Machine learning approaches which do this routinely such as embodied in the SOAR architecture are discussed later in the paper.

Requirement 15: " to predict events in the future or estimate future status".
EON does provide projections to allow comparison such as for the T-Helper implementation of Eon for AIDS - what is the situation 172 hours after symptomatic treatment.

Requirement 16: "ability to evaluate its own performance and improve".
EON does not have such ability.

3. Deep Blue's Brand of Intelligence and Grounding in the Interactionist Perspective

Chess was long seen as an extreme test of human intelligence and an excellent domain for IS [Levinson, 1991]. Chess performance is easy to monitor because success and skill categories are well defined. Studies of experts have been conducted to construct cognitive models which have been more broadly applied (uncertainty management and problem space pruning for example). However, as long as a dozen years ago computer success at chess was largely based on brute force computation using alpha-beta minimax search with selective extensions IS [Levinson, 1991], rather than elegant knowledge structures or complex processing strategies - the intelligent parts of a cognitive model. This was necessary to achieve effective time performance - conventional AI techniques were too slow for real-time response and chess is very much a time bound game. In the late 90s Deep Blue achieved its victory and it's processing capabilities are well known but the victory raises some interesting issues. Foremost is, do we consider Deep Blue intelligent? Behaviorally the answer has to be yes. If we take strictly behavioral views of intelligence in chess we may list the behavioral pattern without making any claim of an agent's cognitive level. Also by a judgement of interacting with its environment it is successful. By performance measures Deep Blue is intelligent, but this seems unsatisfactory on several other levels. It is grounded in the main chess objects and how they behave, but this is trivial, a game of simple rules. Deep Blue doesn't match up against the Messina et al [2001] criteria. It has minimal sensing capabilities, no commonsense knowledge to speak of, no ability to fill in knowledge. An interesting sidelight is that the IBM team found that while chess suggestions from experts were useful, they could not always be relied upon to aid Deep Blue's evaluation function - the essential process. What the team wound up doing is to create a "knowledge-free" machine using just available on-line chess databases to give the system a statistical experience base. That is, the final knowledge base had learned via working through a lifetime of chess games - essentially an grounding in chess reality, leading to its expertise, but the resulting knowledge base lacks the more abstract knowledge such as is often assumed underlies intelligence.

Now it is true that we typically are not talking about ISes with the full range of human ability, but in many cases we are talking about sensori-motor capacity and the ability to distinguish things in the world. Broader world knowledge is more typically learned by robots as summarized by Harnad [1993]in his discussion of a revised test of intelligence he calls the Total Turing Test (TTT):

> Well, in the case of the Turing Test (TT), there *was* more we could ask for empirically, for human behavioral capacity includes a lot more than just pen-pal (symbolic) interactions. There is all of our sensori-motor capacity to discriminate, recognize, identify, manipulate and describe the objects, events and states of affairs in the world we live in (the same objects, events and states of affairs, by the way, that our thoughts happen to be about). Let us call this further behavioral capacity our *robotic* capacity. Passing the TTT would then require indistinguishability in both symbolic and robotic capacity.

We can see this direction as also having been taken by real-time robots to handle problems such as:

- Symbol grounding to the real world (easy in chess, but not elsewhere)
- RT signal interpretation and planning under time constraints
- Situatedness issues - how is behavior adjusted to dynamic situations?

Subsumption architectures have been one attempt around these. [Brooks 1986] The main assumptions behind these attempts include:

- No attempt to construct an full, central symbolic model of the environment
- Behaviors are not controlled by a central executive looking at master plan, but may have a network of behaviors that may excite or inhibit each other.
- Sensor interpretation, planning and execution are not separated. Rather they are organized around modular competencies.
- Complex behavior is not programmed in but emerges from dynamic interactions between the environment and the component behaviors.

As Harnad [1993] says,

> Real transduction is in fact *essential* to TTT capacity. A computational simulation of transduction cannot get from real objects to either robotic performance or symbolic performance (not to mention that motor interaction with real objects also requires the output counterparts of transducers: effectors). This is the requisite nonarbitrary argument for the special status of transduction that we did *not* have in the case of parallelism (or silicon). In addition, there are other things to recommend transduction as an essential component in implementing cognition. First, most of the real brain is either doing sensory transduction or analog extensions of it: As one moves in from the sensory surfaces to their multiple analogs deeper and deeper in the brain, one eventually reaches the motor analogs, until finally one finds oneself out at the motor periphery. If one removed all this sensorimotor equipment, very little of the brain would be left, and certainly not some homuncular computational core-in-a-vat that all this transduction was input *to*. No, to a great extent we *are* our sensorimotor transducers and their activities, rather than being their ghostly computational executives.

There are now examples of unified robotic systems that include a commitment to symbolic manipulation as well as sensory transduction and organized motor responses which might satisfy the TTT, should we want to engage in it. One classic one is ICARUS [Langley et all 1991], which is made up of the standard 3 major components. However, architecturally the sensory buffer proves input to a mapper to the conceptual level, there is a reactive action planner to identify appropriate actions for world situations and there is a mapper of the action to an action "scheme" that drives the effectors that connect to the motor buffer.

The innovation here is that all 3 components use a similar representation (hierarchical probabilistic concepts) and reasoning is driven by a set of heuristics on the classifications in the hierarchy. Thus we have an index of world objects from perception, plans and more schemes each of which is defined by attribute-value pairs with a conditional probability of an attribute having a particular value given membership in a particular class. As noted by van de Velde [1995] by having consistent representation we get natural integration between perception, planning and action. Learning in inherent in the architecture since an instance is sorted down a hierarchy by class selection[2]. When a class is selected the probability distribution of attributes is updated using Classit's incremental function [Gennari et al 1989] or a

[2] One might use other approaches like Grossberg's adaptive resonance theory (ART) to organize clusters as a variety of levels and use the vigilance parameter to adjust the degree of clustering.

new class is created. This integration provides a natural symbol/concept grounding, which is not built in by a knowledge engineer. There need not be world view, rather, like reactive robots success is built through interacting with the world. It goes two steps better than reactive robots in that it creates:

1. manipuable concepts for planning as it goes along.
2. builds coordinated/coupled processes that are closed as well as defined by their organization and the action dynamics these processes imply.

This is also fundamentally different from the traditional I-P-O architecture and the separation between perception and action is largely gone. Both are reactions to environmental changes and employ a planning cognition to preserve or reach some state. Thus, with such an architecture I could speak of a motor performance with plans and percepts underlying it as an integrated thing. More importantly, I might expect the IS to be able to justify its actions by means of a trace of the activated concepts and attribute values. It is a step, but not necessarily as large a step, as we have with Deep Blue to imagine a robust performance over a range of domains.

4. Goal- Oriented Agent Intelligence with Beliefs, Desires and Intentions

In our final dimension I take a large step towards an intelligent performance that might meet some of the judgmental characteristics involved in patient safety. As previously noted categories of patient safety involve a judgement of the intentionality to do wrong. Like the Turing test it implicitly rests on human ability to predict and understand the behavior of "others" in complex interactions. This may lead to misjudging the intelligence of a system. As Hayes and Ford [1995] argue, the Turing Test is fundamentally flawed for two reasons: it is a basically poor experimental design, and it tests for the wrong thing. It is the wrong design because while it seems "unashamedly behavioristic and operationalistic", yet it is based on hidden assumptions rising from on our naïve psychology. Similarity to human behavior is just not a sensible criterion for intelligence. Our social experience provides an implicit, observer bias to assign mentality and intentions to the system in a test and many would argue that typical human use reasoning techniques haven't found their way into typical intelligent systems. E.g., humans use extremely complicated, temporally extended mental images and associated planned intentions to reason. It is the goal of the final dimension to build in such capabilities so that such judgements of IS could be justified.

In the prior section we briefly discussed goals within the ICARUS architecture, but the dynamics of these was not detailed. There is a large body of work to actively incorporate such planning about goals and belief abilities into ISes/intelligent agent architecture as more than reactive systems. ICARUS fits into this class of "cognitive architectures" as does SOAR [Newell, 1990]. SOAR, like ICARUS is interactionist in the sense that the task environment determines the possible structure of problem spaces. It is goal oriented in that problem solving is built around control knowledge that selects goals and sub-goals as it searches a problem state. To understand its behavior we have to look at its functional architecture and ontological commitments to knowledge and goals. Globally problem solving involves search, which is controlled by a context tree which might consist of a 4-tuple of: goal, problem space, state and operator. At any instant we may loosely say that this 4-tuple object is what such an IS actively "knows". The cognitive decision cycle (shown in Figure 3) consists of two phases to manages the context tree by determining what slot attributes should be changes. In the elaboration phase long-term knowledge is represented by production rules fire and those that fit the situational pattern fire in parallel until now more changes in the 4-tuple object occur. In this process new "preferences"[3] for a part of the 4-tuple context object may arise. In the second phase preferences are evaluated via a decision procedure. The result is a new contextual object better than any others. If such an object fails based on preferences one of fours types impasse is reached - tie. No-change, reject impasse or conflict impasse. All impasses are solved by the same goal- search process used in the cycle. Thus the system has a unified approach to problem solving around goal -- based learning that uses environmental results expressed in the problem statement and state as a factor.

SOAR is a pioneering effort which continues, but beside being goal driven we may also introspect about intents. Having a system aware of its performance was listed as an IS feature by Messina et al [2001]. SOAR has a step in that direction, but we wouldn't be comfortable speaking of its intention to "monitor itself". Such intentions built into a system might satisfy professional guidelines for patient safety.

Intentions have been added to agent architectures based on Bratman's [1987] theory of human, rational behavior,

[3] Preference types are fixed simply at elements such as feasibility, exclusivity, desirability, necessity (require, prohibit), termination for desirability of alternative objects occupying slots in the context tree.

Comparison Methodology for Robotic Operator Control Units

Robert Bialczak Ph.D., Jonathan Nida, Brian Pettitt
Technology Service Associates, Inc.

Michelle Kalphat
US Army Simulation, Training, and Instrumentation Command

ABSTRACT

Operator controls and interfaces for military unmanned systems are being developed and used in both laboratory and field research. These controls are considered to be an efficient method of controlling unmanned systems in tactical settings and scenarios. Performance comparisons of different controls and interfaces in differing tactical environments need further study. This paper proposes a methodology of evaluating two selected controls in a laboratory experiment.

The Advanced Robotics Simulations STO research being conducted at the Technology Development Center (TDC) at STRICOM and The Mounted Maneuver Battlespace Laboratory (MMBL) in Fort Knox, Kentucky is using two different control systems. Both are based upon the OneSAF Testbed Baseline (OTB). The first is a version of the OTB that has been extensively modified. Currently it is not instrumented and cannot log the user's actions to the button press level. Instrumenting the SAF would be a relatively straightforward exercise. The second, known as the Operator Control Unit (OCU), has been developed for Future Combat system Experimentation, is built on top of the OTB. The OCU has been instrumented to facilitate analysis. The OCU has the ability to send DIS (Distributed Interactive Simulation) PDUs (Package Data Units) to a logger.

Once fully instrumented, the two controllers would then be run in an exercise. The TDC and the MMBL both have capabilities where 20-30 OCUs and a comparable number of OTBSAF control stations could be utilized. Using the information from varying FCS scenarios an exercise will allow us the ability to analyze the user's actions and the corresponding results. Some of the questions to be examined include:

1) Compare survivability of robots controlled from different stations
2) Compare time to execute commands
3) Compare steps necessary to execute commands
4) When an interesting event happened (e.g. firefights, robot destroyed, etc.), and what actions preceded the event

This research would allow the user community to evaluate not only the effectiveness of the control stations, but in the future evaluate the effectiveness of the man-robot units and interactions.

KEYWORDS: Robot, Control, Compare, Evaluate

1. MEASURING ROBOTIC CONTROLS

There are many ways to measure human to robot controls in tactical scenarios. First you must decide the degree of fidelity for the robotic vehicles and their environment. This ranges from complete immersion in a real world scenario (the robot is in the field in an actual scenario e.g. war) to total simulation (the robot and its environment are modeled by computers.) The most accurate method would be to always test in real situations. Unfortunately that is prohibitive across numerous domains. The complexity of this is also intensified because many robotic systems are still in the design phase. For the purposes of this paper and this experiment we choose to simulate both the robotic vehicle and its environment with a modified version of OTBSAF and a under development experimental FCS OCU. This simulation tool is accurate enough in most battlefield scenarios. We will strive to make the measurements as objective as possible. Only certain aspects of the robotic controls will be measured. One problem that will not be addressed is the level of operator training. Obviously, a poorly trained operator will limit the use of the control unit, so we will assume that all the operators have been sufficiently trained on their control unit.

2. TEST FACILITY

The STRICOM TDC and the Mounted Maneuver Battlespace Laboratory in Fort Knox, Kentucky is the proposed facilities to conduct these experiments. Currently, both facilities have two ways of controlling simulated robotic vehicles at their respective sites – the OTBSAF controller and the FCS Operator Control Unit. Data will be collected with a logger which will record all Distributed Interactive Simulation messages including experimental Package Data Units for button presses from either controller and any communications between the Operator Control Unit and robotic platforms.

2.1 OTBSAF Controller

The first is with an extensively modified version of OTBSAF 1.0. The mechanism for controlling vehicles with this tool has not been changed from the OTBSAF 1.0 baseline. Feedback from controlled vehicles is through a two dimensional tactical map which displays friendly and other known vehicles' status. This version of OTBSAF has not been fully instrumented for button presses, but doing so will be necessary before it can be used for this exercise.

2.2 FCS Operator Control Unit

The second way for controlling robots is with the Operator Control Unit developed for the Mounted Maneuver Battle Laboratory. The Operator Control Unit is a further modified version of OTBSAF 1.0. This tool was designed to simplify many of the user's choices during combat, but still allow flexibility in the system. The Operator Control Unit was also designed to control all robotic entities in the Future Combat System. Feedback from controlled vehicles is through a tactical map as before and with realistic images from sensors on the robotic platform. There have been exercises with more than twenty Operator Control Units in use at the same time. It has been completely instrumented down to the button press level.

3. EXERCISE QUESTIONS

The experiment to be conducted at the TDC and Mounted Maneuver Battlespace Laboratory facilities is proposed to be on up to thirty OCUs and a comparable number of OTBSAF control stations. The same scenarios will be performed with only the robotic control station being different at the start. After the selected scenarios have been completed the information collected by the data logger can be analyzed.

3.1 Mission Success

The most basic question to ask is which robotic controller had the highest success rate for its robots completing their assigned missions.

This may be a difficult problem to solve because of the varying mission types and characteristics of the robots. Scout vehicles may be evaluated on the percentage of enemy vehicles identified or the stealthy ness of how they conducted their mission. Direct fire robots may be evaluated on the amount of enemy vehicles disabled and or destroyed. The mission success for other types of robots (re-supply, engineer) can be assessed in comparable manner.

3.2 Platform Survivability

The next evaluation criteria are which robotic controller allow for the highest survival rate for its robots. The meaning of "survival" must be discussed. Does survival mean the robot is 100% functional or that it still has some use to the controller? A control station that consistently has more platforms completely functional at the end of the exercise must be given more credit than one that has more robots partially disabled (movement impaired, sensors destroyed, etc.)

3.3 Executing Commands

Several ways will be used to evaluate the tasking of vehicles from the control stations. Since the control stations will be completely instrumented, either the time it takes or the number of steps required to complete the command will be captured. Another interesting question currently being posed is what will the number of platforms destroyed be while the user was trying to do other tasks or had their attention diverted to another task and was it significantly different between types of control stations.

4. KNOWLEDGE GAINED

If the appropriate scenarios are conducted and the correct questions are asked and evaluated, the effectiveness of the robotic control requirements can be assessed and evaluated. This method is not limited to the OTBSAF control station and the Operator Control Unit. Other control stations can be integrated into this environment. The Demo III Operator Control Unit was previously used in the Mounted Maneuver Battlespace Laboratory simulated environment to train soldiers before controlling actual robots in the field. An effort to make the messages of the current Operator Control Unit both JAUGS and 4D-RCS compliant is being discussed. This would enable integration of another JAUGS or 4D-RCS compliant controllers with minimal effort. In the future with a consistent method to evaluate measure and asses robotic control stations, the encompassing problem of evaluating the man-robot unit and their interactions can be addressed.

PART I
RESEARCH PAPERS

Plenary Lecture 3 - 2

An Integrated Modeling and Simulation Methodology for Intelligent Systems Design and Testing

Xiaolin Hu and Bernard P. Zeigler
Arizona Center for Integrative Modeling and Simulation
The University of Arizona
Tucson, AZ, USA 85721
www.acims.arizona.edu

ABSTRACT

Model continuity refers to the ability to use the same model of a system throughout its design phases. For intelligent systems, we can restrict such continuity to the intelligent control components, and more specifically, the models that implement the system's decision making. behavior. In this paper, we show how a modeling and simulation environment, based on the DEVS formalism,, can support model continuity in the design of intelligent systems. For robotic systems, such continuity allows design and testing of the same control logic model through the phases including logical simulation, real-time simulation and actual execution.

KEYWORDS: *Model Continuity, Modeling, Simulation, Experimental Frame, Real Time Systems, Intelligent Systems, DEVS*

1. INTRODUCTION

One criterion for intelligence is the ability to make decisions in a timely manner. Certainly for systems expected to interact with the real world, such as robotic systems, real time constraints play a major role, although they may vary in stringency for different behaviors. With the rapid advance in processor speed, memory capacity, sensors and actuators, and dramatic increases in network technology, intelligence has a natural association with distributed systems, as exemplified by multi-agent systems. Unfortunately, the lack of good design methods and support tools has made software development for intelligent systems a bottleneck. To address the importance and complexity of real time software development, academic and commercial tool developers have proposed various real time software models and methods that represent different emphases on this problem. However, so far none of them fits very well to support real time software from a systematic way. A formal methodology is needed for real-time software development [1, 2]. The method should support software development for intelligent systems including designing, testing and execution in a systematic way, with a framework to integrate a system's behavior, structure and timeliness together.

In this paper, we describe an approach to develop real time software for intelligent systems. This approach is based on DEVS modeling and simulation framework [3]. Corresponding to the general "Design—Test—Execute" development procedure, our approach provides a "Modeling—Simulation—Execution" methodology which includes several stages to develop real time software. In the modeling stage, Atomic and Coupled models are built to capture a system's behavioral and structural properties. In the simulation stage, a series of simulators is chosen to simulate and test model's behavior in an incremental fashion step. In the execution stage, the verified model is executed by real-time execution engine. It is important to point out that during the whole process, we maintain model continuity because the same model that has been designed will be simulated and then executed. For distributed systems, this continuity also means the coupling among the models is maintained even though the models are executed in a distributed environment. We believe keeping model's continuity is an efficient way to manage software's complexity and consistency. With model's continuity, we are confident that the system in operation is the system we wanted to design and will carry out the functions as tested by simulation.

This paper will start with the description of the methodology for a stand-alone real time system. Then it will scale up to distributed real time systems. For both systems, step-wise simulation methods are provided to simulate and exercise the model under test. Finally, we describe how experimental frame, a more general testing environment, can be integrated to test the model of interest while still preserving model continuity.

2. MODELING, SIMULATION AND MODEL CONTINUITY

Intelligent real time systems monitor, respond to, or control, an external environment. This environment is connected to the digital logic through sensors, actuators, and other input-output interfaces [4]. A real time system from this point of view consists of sensors, actuators and the real time control and information-processing unit. For simplicity, we will call this last one the control model. The sensors get inputs from the real environment and feed them to the control model. The actuators get commands from the control model and perform corresponding actions to affect the real environment. The control unit processes the input from sensors and makes decisions based on its control logic. Depending on the complexity of the system, the control model could have only one central component or it could have multiple parallel-processing subcomponents, which in turn may have their own sub-control units.

Once we establish this view of a real time system as shown on the left side of Figure 1, we can model it easily. In our approach, sensors and actuators are modeled as DEVS Activities, which is a concept introduced by RT-DEVS for real time system specification [5]. A DEVS Activity can be any kind of computer task. However, in the context of this paper, we only consider the sensor/actuator Activities. The control unit is modeled as a control model which might has a set of subcomponents. These subcomponents are coupled together so they can communicate and cooperate. With this approach, the control model acts as the brain to process data and make decisions. It could be a simple Atomic model or a complex hierarchical coupled model. Sensor/actuator Activities act as hardware interfaces providing a set of APIs for the control model to use. They are essentially hardware drivers for sensors and actuators. How to define an Activity and its APIs is dependent on how the designer delineates the "Control Model—Activity" boundary. For example, we can model a sensor module that may have its own control logic as a sensor Activity. Or we can also include that part of logic into our control model and only model the sensor hardware as an Activity. The clear separation between control model and activity's functions makes it possible for the designer to focus on his design interest. In the context of intelligent real time systems, the control logic is typically very complex, as the system usually operates in a dynamic, uncertain or even hostile environment. As such, the control model is the main interest of design and testing. In our approach, simulation methods are applied to test the correctness and efficiency of this model. Tne continuity of this model is also emphasized during the whole process of the methodology.

Before simulating the control model, we need to model the real environment as an environment model. This environment model is a reflection of how the real environment affects or is affected by the system under design. Meanwhile, a "simulated" sensor/actuator hardware interface is also needed for the control model to talk to the environment model. This is why we introduce the SimActivity concept. In contrast to an Activity, which drives real hardware and is actually being executed, a SimActivity imitates an Activity's interface/behavior and

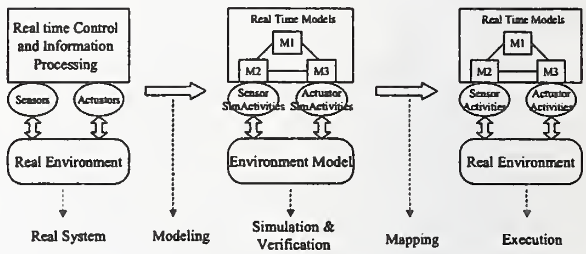


Figure 1: Modeling, Simulation and Execution of Non-distributed Real Time System

is only used during simulation. A sensor SimActivity gets input from the environment model just as a sensor Activity gets input from the real environment. An actuator SimActivity does similar things as an actuator Activity too. Note that it is important for an Activity and its SimActivity to have the same interfaces, which are used by the control model in both simulation and real execution. By imposing this restriction, the control model can be kept unchanged in the transition from simulation to execution (it interacts with the environment model and real environment using the same interfaces). Thus, model continuity is achieved.

As shown in the center of Figure 1, in the modeling stage, a simulated system is developed based on the real system. With this system, different simulation strategies can be applied to validate the control model. In DEVS, there is a clear separation between a model and its simulators, which gives us the flexibility to choose different simulators to simulate the same model. These simulators include fast-mode simulator, real-time simulator and distributed simulators. With these simulators, a model can be simulated and tested incrementally before its real execution. During the simulation stage, employing fast-mode (or logical time) simulators, if we find the simulated result is not what we expected, the model can be revised and then re-simulated. This "modeling-simulation-revising" cycle repeates until we are satisfied with simulation result or nothing more can be learned in the simulation stage. A more detailed description of how to choose and use different simulators is given in the next section.

After the model is validated through simulations, it will be mapped to the real hardware for execution. For a non-distributed application, this mapping is the "Activity Mapping" to associate the sensor/actuator Activities to the corresponding sensor/actuators hardware. For a distributed application, an extra "Model Mapping" is needed to map a set of cooperative models to a set of networked nodes. By associating the models and Activities to their corresponding hardware, the system can be executed in a real environment. In execution, the control logic is governed by the control model, which has been validated in the step-wise simulation. If true model continuity from simulation to execution has been achieved, this control model will carry out the control logic just the same as it did when simulated. In practice, one may not be able to completely replicate the real environment in the environment model, and there will be potential for design problems to surface in real execution. When this happens, re-iteration through the stages can be more easily achieved with the model continuity approach.

3. STEP-WISE SIMULATION AND TESTING

Simulation technology has been widely applied to help to design and test real time systems. This technology provides a valuable tool for engineers to test and understand the system under design. When the complexity of a problem is too large to allow an analytical solution, simulation is the only option to investigate system configurations or operational modes prior to the implementation in the field. In this section, we will show how different simulation methods can be applied to incrementally simulate and test a stand-alone system. Simulations for distributed systems will be shown in section 5.

As shown in step 1 of Figure 2, for a stand-alone system, three different simulation steps can be applied to test the model. They are fast-mode simulation, real-time simulation and hardware-in-the-loop simulation. These simulation methods apply different simulation configurations to test different aspects of the model under test.

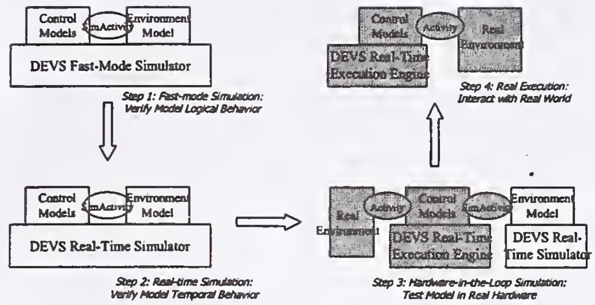


Figure 2: Step-wise Simulations of Non-distributed Real Time System

In fast-mode simulation, the control model is configured to talk to the environment model through sensor/actuator SimActivities. These models stay in one computer and a DEVS fast-mode simulator is chosen to simulate them. In fast-mode simulation, the flow of time is logical (not connected to a wall-clock). So, a fast-mode simulator generates simulation results as fast as it can. Based on these results, we can analyze the data to see if the system under test fulfills the logical behavior as desired.

Just as fast-mode simulation verifies a model's logical behavior, real-time simulation verifies model's temporal behavior. In real time simulation, the model's setting is the same as in fast-mode simulation. However, the fast-mode simulator is replaced by a real-time simulator, which executes the model at the same speed as a real world clock. Since the simulation runs in real time, we can test a model's temporal behavior such as checking if critical deadlines can be met.

In fast-mode and real-time simulations, the model under test and the simulators reside in one computer. This computer is not the same computer as the one in which the model will actually be executed. Instead, a simulated environment is provided. However, not all components in a complex system can be modeled in adequate detail in computer simulation. Sometimes, the executing hardware can have significant impact on haw well a model's functions can be carried out. For example, processor speed and memory capacity are two typical factors that can affect the performance of an execution. Thus, to make sure that the control model, having been validated in fast-mode and in real-time simulation, also can execute correctly in the real hardware, we adopt the hardware-in-the-loop (HIL) simulation [6,7]. As shown in step 3 of Figure 2, in HIL simulation, the environment model is simulated by a DEVS real time simulator on one computer. The control model under test is executed by DEVS real-time execution engine on the real hardware. This DEVS real-time execution engine is a stripped-down version of DEVS real-time simulator. It provides a compact and high-performance runtime environment to execute DEVS models. In HIL simulation, the model under test interacts with the environment model through SimActivities. These SimActivities act as simulated sensors or actuators. Real sensors or actuators can also be included into HIL simulation by using sensor/actuator Activities. The decision of which sensor/actuator will be real hardware and which sensor/actuator will be simulated SimActivities is dependent on the test engineer's testing objectives. With different testing objectives, different combinations of real sensor/actuators and simulated sensor/actuators can be chosen to conduct an exhaustive test of the control model. Notice that in HIL simulation, as the control model and environment model stay on different computers, a bi-directional connection must be established between the two computers. We use LAN connection based on TCP/IP protocol because it is widely used in industry, can sustain high-speed data transfer, and very portable. This connection is taken care of by DEVS real-time simulator and execution engine so it is transparent to the model.

Once we passed hardware-in-the-loop simulation, we are ready to leave the simulation stage for execution stage. As shown in step 4 of Figure 2, in real execution, DEVS real-time execution engine executes the control model. There is no environment model because the control model will interact with the real environment through the sensor/actuator Activities.

4. DISTRIBUTED REAL TIME SYSTEMS

With the advance of network technology, distributed real time systems are playing more and more important roles. Figure 3 shows an example distributed system with three nodes. Generally speaking, a distributed real time system consists of a set of subsystems. Like a stand-along system, each subsystem has its own control and information processing unit and it interacts with the real environment through Sensor/Actuators. However, these subsystems are not "along". They are physically connected by network and logically they talk to each other and cooperate to finish a common task. Distributed real time systems are much harder to designed and tested because one subsystem's behaviors may affect one or all of other subsystems. These subsystems influence each other not only by explicit communications, but also by implicit environment change as they all share the same environment. For example, in Figure 3, if Node 1 changes the environment through its actuators, this change will be seen by the sensors of Node 2, thus affects Node 2's decision making. With this kind of influence property, it's not practical to design and test each subsystem separately and then put them together. Instead, the system as a whole needs to be designed and tested.

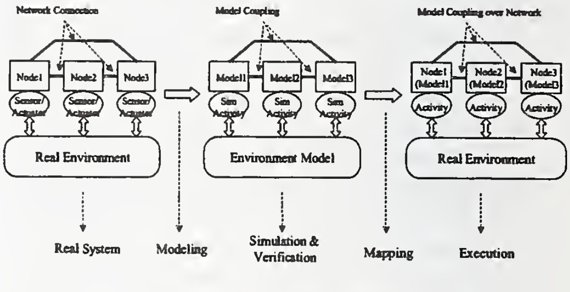


Figure 3: Modeling, Simulation and Execution of Distributed Real Time System

In our approach, a distributed real time system is modeled as a coupled model. This coupled model consists several subcomponents. Each subcomponent is corresponding to a subsystem of the distributed real time system. As described in section 2, these subsystems are also modeled as DEVS models, which consist of control model and sensor/actuator Activities. The control model of each subsystem interacts with the real world through sensor/actuator Activities. These subsystem models are coupled together (by connect one model's output port to another model's input port) so they can communicate. The coupling among the models is corresponding to the connection among the subsystems in the real world.

To test the models of distributed real time systems, simulation methods are applied in our approach. For the purpose of simulation, environment model and sensor/actuator SimActivities are developed to simulate the real environment and sensor/actuator Activities. An Activity and its corresponding SimActivity share the same interfaces so the model using them can keep unchanged from simulation stage to execution stage. Different simulation methods can be applied to simulate and test the models incrementally. These simulation methods include centralized fast-mode and real-time simulation, distributed real-time simulation and hardware-in-the-loop (HIL) simulation. A more detailed description will be given in the next section. Note that each subcomponent can also be tested/simulated independently because DEVS has a well-defined concept of system modularity.

After the models are validated by simulations, they are mapped to the real hardware for execution. Similar to a stand-alone system, each subsystem needs to conduct an "Activity Mapping" to associate the sensor/actuator Activities to the corresponding sensor/actuator hardware. In addition, as the models are actually executed on different network computers, a "Model Mapping" is needed to map the models to their corresponding host computers. These computers are physically connected by the network and they execute the models that are logically coupled together by DEVS coupling. To govern this mapping, a prototype Model Mapping Specification has been developed, which will map the models to their network nodes, while maintaining the coupling among them. As such, model continuity for distributed real time systems means not only the control model of each subsystem remain unchanged but also the coupling among the component models is maintained from the simulation to distributed execution.

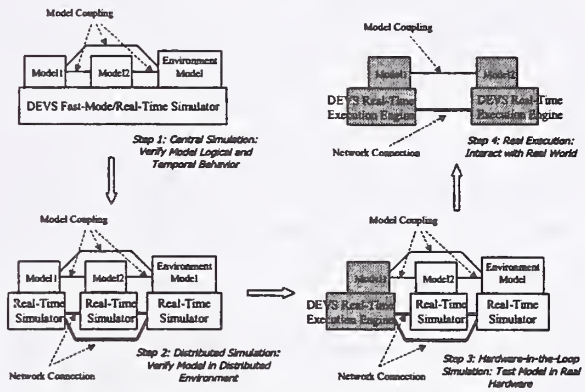
In real execution, the control model of each subsystem makes decisions based on its control logic. It interacts with the real environment through sensor/actuator Activities. If a model sends out a message, based on the coupling, this message will be sent across the network and put to another model's input port. Again, with model continuity, all the subsystems will work and cooperate as were simulated.

5. SIMULATION AND TESTING OF DISTRIBUTED REAL TIME SYSTEMS

Distributed real time system is inheritly complex because the functions of the system are carried out by distributed computers over network. With our approach to model the whole system as a large coupled model, this model can be simulated and tested in our simulation framework. To enable simulation, environment model and sensor/actuator SimActivities are developed to simulate the real environment and sensor/actuator hardware. In this section, three different simulation methods are shown to give a step-wise simulation and testing of the models. These methods are central simulation, distributed simulation and hardware-in-the-loop simulation. To help to understand these methods, an example distributed real time system with two network nodes (two component models) is shown in Figure 4.

The first step is central simulation. In central simulation, the two models and environment model are all in one computer. Fast-mode simulator and real-time simulator are chosen to simulate and test the model respectively. As fast model simulation verifies system's logic behavior, real time simulation verifies system's temporal behavior.

As central simulation test models' logic and temporal behavior in one computer, it doesn't consider the network effect such as network delay. There are two ways to take account of this network factor. One way is to model the network and add the network model into central simulation. Another way is to run simulation over the real network. We adopt the second way to conduct distributed simulation of the system. In order to conduct a meaningful testing, the network the simulation is running should be the same or at least similar to the network the model will be really executed. As shown in figure 4, in distributed simulation, two models stay on two different computers. The environment model may stay on another computer or on the same computer as one of the models. The coupling between these computers remains the same, but it happens across the network. All of these models are simulated by real time simulators. These real time



simulators take care of the underline network

Figure 4: Simulation of Distributed Real Time System

communication so it is transparent to the model. As such, there is not need to change the model for network communication.

In distributed simulation, the real network is included so the system is simulated and tested over the real network. To further this test, real hardware the model will be executed can also be included into our simulation. This is the hardware-in-the-loop (HIL) simulation. In HIL simulation for distributed real time systems, one or more models can be distributed to their hardware to be simulated and tested. In the example of Figure 4, Model 1 along with its real-time execution engine stays on the real hardware. Model 2, environment model and their real time simulators stay on other computers. These models still keep the same coupling. However, the model on real hardware may use some or all of its sensor/actuator hardware to interact with the real world. Similar to the description in section 3, different configuration can be applied to test different aspects of the model.

After all these simulations, we have confidence that the distributed system will operate as we simulated. Then the models are mapped to the real hardware for execution. In real execution, DEVS real-time execution engine executes the model and take care of the underline network communication. The environment model is gone, as all models interact with the real environment.

6. EMPLOYING EXPERIMENTAL FRAMES FOR TESTING

In previous sections, we have shown that simulation methods can be applied to test distributed or non-distributed real time systems in an incremental fashion. . We have discussed a testing methodology that consists of an environment model and SimActivities in which control logic can be tested. Since such testing mainly focus on the interaction between the environment model and control model, a more general testing environment can be developed using the concept of experiment frame. An experimental frame is a specification of the conditions under which a system is observed or experimented with [3]. A typical experimental frame has three types of components: *generator*, which stimulates the system under investigation in a known, desired fashion; *acceptor*, which monitors an experiment to see that desired conditions are met; and *transducer*, which observes and analyzes the system outputs. In the context of real time software design and test, an experimental frame acts as a test module to serve the functions of a test event generator, test monitor and performance analyzer. The real environment in which the application is embedded is usually modeled and included in the experimental frame. A related example can be found in [8].

With the experimental frame concept, the example shown in Figure 4 to test a distributed system can be generalized as shown in Figure 5. Here, experimental frame replaces the environment model to provide a more general and powerful testing environment. The environment model is still needed inside the experimental frame to interact with control model. However, more special *generators* can also be added into the experimental frame to provide special case test. Notice that with experimental frames, not only can the control logic be tested and validated, but also the performance of the model, such as average response time, can also be measured by using a *transducer*. Moreover attributes of intelligent behavior can be captured through specialized experimental frames and tested in the various phases of development.

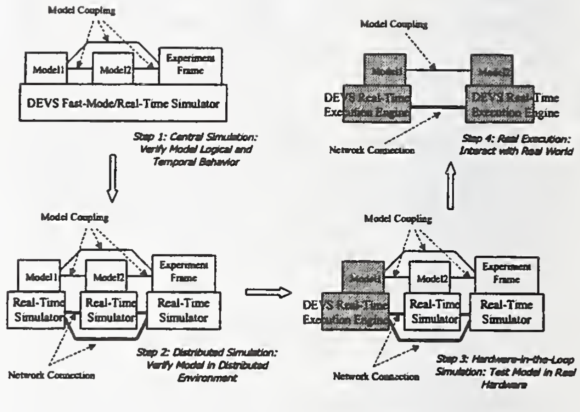


Figure 5: Testing of Distributed Real Time System Using Experimental Frame

In order to maintain model continuity, special attention has to be paid when introducing a test module to conduct testing. For example, in section 2 and 4, SimActivity has been introduced and we require that a SimActivity should have the same interfaces as an Activity. Similar restrictions are also needed when experimental frames are integrated into the system for model testing so that model continuity can be maintained.

7. CONCLUSION

Separation of models and simulators as distinct, though interacting, elements, supports model continuity. This means that the same model may be handled by different simulators appropriate to the design, testing, and execution phases of intelligent system design. Continuity of the control logic model is particularly important for design of real time, distributed intelligent systems, whose complexity would otherwise overwhelm the designers. Modeling and simulation environments, based on the DEVS formalism, can support such model continuity. An example in robotic system design has been developed and will be discussed in future papers.

8. REFERENCES

[1] Lee, E.A., "What's Ahead for Embedded Software" *IEEE Computer*, Volume: 33 Issue: 7, Sep 2000.

[2] Pimentel, A.D.; Hertzbetger, L.O.; Lieverse, P.; van der Wolf, P.; Deprettere, E.E. "Exploring embedded-systems architectures with Artemis" *IEEE Computer* Volume: 34 Issue: 11, Nov. 2001

[3] Zeigler, B.P., T.G. Kim, and H. Praehofer, *Theory of Modeling and Simulation. 2 ed. 2000*, New York, NY: Academic Press

[4] Slan C. Shaw, *Real-time Systems and Software*, 2001, John Wiley & Sons

[5] J.S. Hong, and T.G. Kim, "Real-time Discrete Event System Specification Formalism for Seamless Real-time Software Development," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 7, pp.355-375, 1997.

[6] Feijun Song; Folleco, A.; An, E., "High fidelity hardware-in-the-loop simulation development for an autonomous underwater vehicle" OCEANS, 2001. MTS/IEEE Conference and Exhibition, Volume: 1 , 2001

[7] Wells, R.B.; Fisher, J.; Ying Zhou; Johnson, B.K.; Kyte, M., "Hardware and software considerations for implementing hardware-in-the-loop traffic simulation " Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE , Volume: 3 , 2001

[8] Schulz, S.; Buchenrieder, K.J.; Rozenblit, J.W. "Multilevel testing for design verification of embedded systems" *IEEE Design & Test of Computers* Volume: 19 Issue: 2 , March-April 2002

PART I
RESEARCH PAPERS

3P1 - Uncertainty in Representation II

1. Multiple Neural Network Model Interpolation
   D. Chin, Johns Hopkins University, Applied Physics Laboratory
   A. Biondo, Johns Hopkins University, Applied Physics Laboratory

2. A Three-Tier Communication and Control Structure for the Distributed Simulation of an Automated Highway System
   R. Maarfi, Tennessee Technological University
   E.L. Brown, Tennessee Technological University
   S. Ramaswamy, Tennessee Technological University

3. Tessellating and Searching in Uncertain State Spaces
   A. Meystel, Drexel University
   A. Bathija, Drexel University

4. Autonomy and Socialization
   K. Bellman, The Aerospace Corporation

Multiple Neural Network Model Interpolation

Daniel C. Chin and Albert C. Biondo
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel Maryland 20723-6099
daniel.chin@jhuapl.edu and albert.biondo@jhuapl.edu
Phone : (240) 228-4956

Abstract

This paper presents an efficient method for extracting a multi-model interpolation function from a nonlinear system. The multi-model interpolation function consists of couple simplified time-varying models in neural-network structure to dynamically approximate the nature of the physical phenomena to be interpolated and extrapolated. The purpose of using the multi-model interpolation function is to perform a real-time approximation. This paper demonstrates the interpolation in a simulated environment, the underwater acoustic transmission loss generated from the NAVY-standard acoustic propagation-loss model ASTRAL, which is not suited to real-time operation. The interpolation includes initial learning period that is on the order of 20 minutes (more or less time depends on the size of the parameter intervals and the complexity of the ocean environment), and the subsequent interpolation speed will be measured in fractions of a second, a several orders-of-magnitude improvement over conventional calculations. In addition, for the example presented here, the interpolation error is within 1% of the actual transmission-loss value in a root-mean-square (RMS) sense.

Keywords: Multi-model Interpolation; Multi-objective SPSA; Nonlinear Interpolator; Neural Network; Nonlinear acoustic wave function.

1 Introduction

This paper develops a model-fitting technique to perform the interpolation and extrapolation of a nonlinear time-varying system. The development is demonstrated on the problem of transmission loss of underwater sound. The technique involves simplified time-varying multiple models, neural networks (NN), and multi-objective simultaneous perturbation stochastic approximation (MSPSA). The simplified models represent the local phenomena that change in time; NN projects the model variations; MSPSA trains the NN-weights. The MSPSA was first introduced in Chin [1] and is based on the simultaneous perturbation stochastic approximation (SPSA) developed by Spall [2]. A collection of applications of NN in adaptive control of nonlinear systems can be found in Ng [3]. The localized multi-model technique has shown accuracy and efficiency in the transmission loss interpolation.

The transmission loss function is highly oscillatory and quite variable. There is no simple representation available to describe the sound wave propagation accurately. The various local medium interactions and reflections give the function its erratic structure. An interpolation method suggested for time-variant systems in Gohberg (Ed.) [4, pp. 153-259] was too complicated and worked only on a single model. In a previous study, a linearized interpolation approach, a simple linear fit between observed data points, was suggested and tried in FY98 Progress Repot to DARPA [5]. · Although the linearized interpolator would save computation time over the actual simulation calculation, the preparation of the base transmission loss curves and the massive amount of data handling ultimately make the linearized interpolation intractable. Also, the resulting interpolation errors were not uniform throughout the parameter space interval desired for the interpolation.

This particular model-fitting design uses two independent neural networks as the base of interpolation. The models are designed to fit the local physical phenomena and the NN's store the model variation information. The interpolator is expected to approximate the sound wave transmission loss accurately within the training area along the transmission pass, therefore the training process should provides NN the intermittent information. There are two ways to provide the intermittent information: 1) from an accurate model representation for the inverse estimation like the one introduced in Chin [1]; or 2) the intermittent observations derived from a base model like the ones discussed here for interpolation. The intermittent observations are accessible in most simulation packages; the utilization allows the model-fitting technique to use less number of simulated transmission passes and to gain more information in preparation of the interpolator.

In comparison with a simple linearized interpolator, the model-fitting technique described here requires longer time per interpolation, but takes orders of magnitude less preparation time. The interpolation time for the model-fitting technique in comparison with the detailed simulation time is negligible. The example presented in this paper shows that using 10 propagation loss curves is enough to train the interpolator for a large portion of the parameter space where interpolation is desired. The base-propagation loss curves for linearized interpolation would use order of magnitude amount more transmission curves to achieve a comparable level of accuracy. The ability to use a few propagation-loss curves to train the NN-weights for accurate interpolation makes the model-fitting technique desirable in planing a real-time simulation-training mission that was questionable for the linearized interpolator.

The NN-weight training procedure also is a very important task for the interpolator; it should consider the matches for intermittent points and the divergent of two different objectives in the two models. Given the variability and oscillatory behavior of the function, the training process also should have some checks and balances. One way to deal with the multiple-objective problem is summing these objectives and forcing them into a single objective algorithm. However, it is then very hard to find the balance among the objectives and the convergence speed, see Chin [1]. The MSPSA algorithm introduced in Chin [1] optimizes the independent model parameter sets (the parameters in one set have no relationship with the parameters in other set) from relevant objectives and is suitable for training the NN-weights here. Also, the parameter dependencies are different for the two models, the algorithm could tailor the minimization procedure to accommodate the differences. In the end, MSPSA used small number of detailed simulation curves to train the interpolation functions, achieved acceptable root-mean-square errors from the original simulation results, and made real-time operation feasible.

2 Underwater Sound Transmission Loss

Follow the principle of underwater sound in Urick [6], the sound propagating through the ocean was described in three physical phenomena:

- Sound spreads while it propagates through the medium in three different ways: spherically, cylindrically, and linearly.
- The medium absorbs sound energy, with the rate of absorption varying with the water temperature and the acoustic frequency.
- Sound signals are also influenced by the reflections from the top and bottom of the ocean water column. This influence is a function of the local bottom bathymetry and composition, as well as the sea surface conditions (wave height).

These three effects also vary with the frequency of the sound signal.

Using two models could describe these three phenomena. One model approximates the energy spreading and absorption because their equations are similar; the other model approximates the reflections. The energy dissipation models dependents on water temperature at the referenced local areas; the reflection model depends on both site structures and range from the sound source.

The transmission losses are represented as a ratio of the sound intensity at a given range, say p nmi, to the intensity of a reference range. If TL represents the transmission loss and T represents the signal intensity, the sub-indexes represent the values evaluated at the reference points, respectively. Then, the relation between the transmission loss and signal intensity are as following:

$$TL_p = 10\log\left(\frac{T_p}{T_0}\right). \qquad (1)$$

The intensity T changes along the transmission pass in a non-linear pattern. For easier to approximate the intensity reduction, the sound propagation (the thick line inside the box on Figure 1) is divided into a fine fixed-interval grid, the grid points are located at unit marks for convenience of data handling, in our study here uses nautical mile marks. The marks 0, p on Fig. 1 are the reference points near source and at receiver. The marks $1, 2, \cdots, k$ are the reference points on the grids.

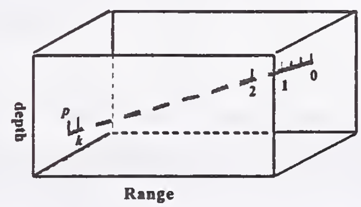


Figure 1: The reference points and sound propagation pass

The total sound transmission loss between source and receiver could be expressed as the accumulation of transmission losses along the pass, as shown in (2); an expansion of Equation (1):

$$TL_p = 10\log\left(\frac{T_p}{T_k}\right) + \left[10\log\left(\frac{T_k}{T_{k-1}}\right) + \cdots + 10\log\left(\frac{T_1}{T_0}\right)\right]. \qquad (2)$$

The examples in Section 5 use the nautical mile as grids and 3-ft as the initial 0 reference points. The receiver mark p is located between grids k and $k+1$ include $k+1$.

At each reference points, $i \in \{0,1,\cdots,k,p\}$, the transmission loss is also the total effect from all three individual physical effects that are represented by two models. Let T_i^E and T_i^R be the signal intensity reduction value due to energy dissipation and reflection between the reference points $i-1$ and i. The total sound wave intensity at the reference point i would be

$$T_i = d\left(T_i^E - T_i^R\right). \tag{3}$$

where d is the distance between interpolating position and reference point $i-1$, in (3) the interpolating position is the position of reference point i. Substitute $T_i\ \forall i$ into (2), and then computes the total transmission loss from source to receiver.

To simulate T_i from detailed non-linear models are both computational and computer I/O intensive operations. Simplification of the data structure and retrieval system are the first couple steps in reducing the computational burden and data handling problems. Because mass amount of data for the water temperature profiles and detailed ocean basin information along the transmission pass are required in computing the transmission losses from the non-linear accurate models. This paper utilizes a pair of NN for so purpose.

The equation for energy dissipation intensity reduction formula T_i^E is a simple constant varying mostly with the water temperature (assuming uniform value within two reference points) and could theoretically have value range between 0 and 3 (not including 0). For generality, the equation is expressed in two degree of freedom among T_i^E's as following:

$$T_i^E = s\left(1 - \frac{r}{2}\right), \tag{4}$$

where s and r are two constants with values between 0 and 1 that will be the output from NN (the outcome could be in expected range for the interpolation area of interests, instead of the theoretical range). There is no simple expression for T_i^R. This paper uses the first order of trigonometric function to represent the energy gain from reflection as following:

$$T_i^R = a\cos\left(\frac{\pi\phi p}{720}\right) + b\sin\left(\frac{\pi\varphi p}{720}\right), \tag{5}$$

where a, b, ϕ, and φ are four coefficients and would be the output from NN, π is the radian constant, $3.14159\cdots$. This equation may be changed due to environmental differences, e.g., with a higher order representation for a more complicated environment.

The neural-networks are designed for tracking the variations of the coefficients used in (4) and (5) among the intensity reduction functions due to environmental change along the transmission pass. The neural-network for the energy dissipation model is a two-hidden layer network with four inputs and two outputs. The number of weights for each of the two-hidden layers is five. The four inputs are frequency, source depth, depth at the initial reference point, and delta range from the interpolation point to the initial reference point. The two outputs denoted by s and r are the spreading factor and absorption rate as defined before. The neural-network for the reflection model is a one-hidden layer network with four inputs and four outputs. The number of weights for the hidden layer should be changed according to the size of the geographical area, the bottom type and the sound frequency; a larger area, more complex bottom types and higher frequencies will use a larger number of the NN-weights. The inputs for this network are frequency, source depth, depth at the initial reference point, and range of the range from source. The outputs for this network are the coefficients of a trigonometric equation.

3 Interpolation Setting

The underwater sound transmission loss can be expressed in a more general term, such as a system y consists of two models denoted by $f(\bullet)$ and $g(\bullet)$, and

$$y = F(f(\bullet), g(\bullet)), \tag{6}$$

where the function F is nonlinear and model f and g are varying with time. The value of y can be accumulated from a sequence of intermediate function values y_i and

$$y = \sum_{i=0}^{p} y_i \tag{7}$$

where y_i is evaluated at the reference point i and $i \in \{0, 1, \cdots, k, p\}$. Reference points 0 and p are located at the boundary points; reference points $\{1, \cdots, k\}$ are located at the internal grid points along the transmission pass. The individual y_i is also a function of $f_i(\bullet)$ and $g_i(\bullet)$ and

$$y_i = F(f_i(\bullet), g_i(\bullet)) \tag{8}$$

Assuming the functions $f_i(\bullet)\, \forall i$ can be approximated by the same function with different coefficients such as the one in (4), likewise for $g_i(\bullet)\, \forall i$ as the one in (5).

Let NN denote a neural network, with NN_f and NN_g the neural networks for models f and g, respectively. Assume $x_{f,i}$ and $x_{g,i}$ are the input terms of NN_f and NN_g at the interval between reference points i-1 and i. Then,

$$\begin{aligned} x_{f,i} &\rightarrow NN_f \rightarrow f_i \\ x_{g,i} &\rightarrow NN_g \rightarrow g_i \end{aligned}, \tag{9}$$

where f_i and g_i are the neural network output parameters associated with the two models at the same interval and will be used as the coefficients of $f_i(\bullet)$ and $g_i(\bullet)$. Then these two functions could be defined as $f_i(x_{f,i} \mid w_f)$ and $g_i(x_{g,i} \mid w_g)$, where w_f, w_g are the weights of NN_f, NN_g. Function $f_i(\bullet)$ is the function of $x_{f,i}$ based on the weight values of NN_f; similarly function $g_i(\bullet)$ is the function of $x_{g,i}$ based on the weight values of NN_g. Let $\hat{\theta}_f, \hat{\theta}_g$ be the estimated variables for w_f, w_g and $\hat{y}$ be the approximation value for y and

$$\hat{y} = \sum_{i=0}^{p} F\left[f_i\left(x_{f,i} \mid \hat{\theta}_f\right), g_i\left(x_{g,i} \mid \hat{\theta}_g\right)\right]. \tag{10}$$

We are trying to minimize $(y - \hat{y})^2$ and combinations of $(y_i - \hat{y}_i)^2$ for all transmission loss curves to find the best fit $\hat{\theta}_f, \hat{\theta}_g$ of w_f, w_g. Then we could use $\hat{y}$ as an interpolation value from the given sets of input $\{x_{f,i}\}$ and $\{x_{g,i}\}$, i.e. the input parameters defined at the reference points along sound transmission pass as they are define in Section 2. This setting may easily be expanded into a system that involves more than two models and may also be used in a control environment.

4 The Training Algorithm

The multiple-objective simultaneous perturbation stochastic approximation (MSPSA) algorithm presented in [1] is used to train the neural-network weights. The algorithm attempts to minimize the sum of the square difference between interpolation values and the computed values over the local intervals, the reference points assumed in the derivation of the equations, as well as the entire data range. The differences are calculated at each computed value, according to the resolution of the data. The minimizations are completed over iterations. Any single iteration consists of many small steps from the individual minimizations that are sequenced one-by-one; let us call the small step a minimization step. The estimates of one minimization-step will be passed to the next step in the sequence as the previous estimates of that step. The estimates from the last step in the sequence will be the estimates of the iteration. The estimates of the iteration will be passed to the first minimization step in the next iteration as the previous estimate of that step. This optimization algorithm assumes the data was generated with a consistency setting, similar environment or limited interference. Simulation data has less of a consistency problem then the real data. Even the inconsistency does exist among the data, the order of the step sequence will not affect the outcome of the estimates, and it just effects the convergence speed. The minimization-step procedure uses the equation stated in [2] as an iteration of the SPSA algorithm, using two statistical perturbation estimates to approximate a gradient that updates the previous estimates.

For convenience, let f represent the energy dissipation model and g represent the reflection model for the underwater sound transmission loss system. The subscript-index i indicate the local ranges along the transmission loss curve. Figure 2 shows the detailed training procedure of "one iteration" as follows:

1) Starting from the initial estimates or the previous iteration estimates, the first step (box 1) is to minimize the local differences between y_i and $\hat{y}_i\left(f_i\left(x_{f,i} \mid \hat{\theta}_f\right) \mid g\left(x_{g,i} \mid \hat{\theta}_g\right)\right)$ on the weights of NN_f while holding the weights of NN_g unchanged.
2) Using the 1st step estimates as the initials, the 2nd step (box 2) minimizes the full range difference between y and $\hat{y}\left(g_i\left(x_{g,i} \mid \hat{\theta}_g\right) \mid f\left(x_{f,i} \mid \hat{\theta}_f\right), \forall i\right)$ on the weights of NN_g while holding the weights of NN_f unchanged.

3) Initializing from the 2nd step estimates, the 3rd step (box 3) minimizes the summation of the square differences between y_i and $\hat{y}_i\left(f\left(x_{f,i} \mid \hat{\theta}_f\right), g\left(x_{g,i} \mid \hat{\theta}_g\right)\right)$ on the weights of NN_f and NN_g for a global fit.

4) Box 4 – indicates repeating steps 1, 2, and 3 for all source-receiver (SR) pairs.

5) Box 5 evaluations the estimates of weights on both NN_f and NN_g. If the estimation error, ε, is greater than $1.20\varepsilon_{min}$, the last known smallest error, then reject the estimates and repeat the previous 4 steps. If the estimation error is within $1.20\varepsilon_{min}$, then update the weights on both NN_f and NN_g and proceed to the next iteration.

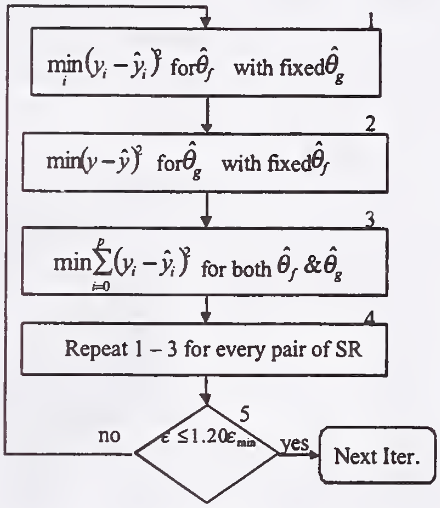


Figure 2 One Iteration of Training Algorithm

5. Example

ASTRAL (the Automated Signal Excess Prediction System (ASEPS) TRAnsmission Loss), is a Navy standard model, included in the Navy's Ocean and Atmospheric Master Library (OAML). OAML is a collection of configuration-controlled models and databases, maintained by the Naval Oceanographic Office (NAVOCEANO). ASTRAL was specifically designed to run rapidly, and is commonly used in real-time simulations because it runs 10 to 1000 times faster than the traditionally more accurate parabolic equation (PE) models, as well as other research models.

ASTRAL is primarily a range-dependent, adiabatic, range-smoothed mode theory model, with additional separate algorithms to model important acoustic features that are not appropriately handled in the primary algorithm. In particular, ASTRAL uses separate algorithms for convergence zone and surface duct propagation [6]. ASTRAL can predict the range averaged transmission loss and vertical angular arrival structure, but only the former quantity is considered here.

The selected model is expected to be run for all propagation calculations required during the Navy simulations in which is used. Therefore, ASTRAL was run for a variety of environmental conditions and operational parameters deemed reasonable for the simulation.

Assuming a simulation for purposes of real-time operator training, the oceanographic environment for this example was located in the Sea of Japan. For simplification, a single set of propagation paths with varying bathymetric details was used in the example. The differences between each of the paths were in the receiver depth, source depth, sound frequency, and transmission range. We assumed that a receiver was placed at certain discrete depths in a 200-ft interval from the surface, with the source placed at various depths between the surface and 1500-ft. Each source and receiver pair was generated using different frequencies, from 20 Hz to 10000 Hz. The total transmission range was 102 nmi; the resolution to which ASTRAL generated was 0.25 nmi.

The receiver depth, source depth, frequency, and transmission range define the parameter space interval desired for the interpolation, or the interpolation area, which also defines the real-time operator training area. In order to have the interpolator work properly; it has to learn the characteristics of the transmission loss from the data computed within the interpolation area. The main criterion for data selection is that the selected data has to contain all the important features in the area. For the time being, we were using trial and error to select 10 source-receiver pairs, each of them at a different frequency. Figure 3 shows the transmission loss surface formed by the 10 selected source-receiver pairs. The Y-axis in the Figure indicates the 10 selected pairs from 1–10; the X-axis indicates the total transmission range from 1 – 102 (102 nmi in 0.25 nmi resolution); and the Z-axis shows the scales of the transmission losses from 50 dB to 100 dB.

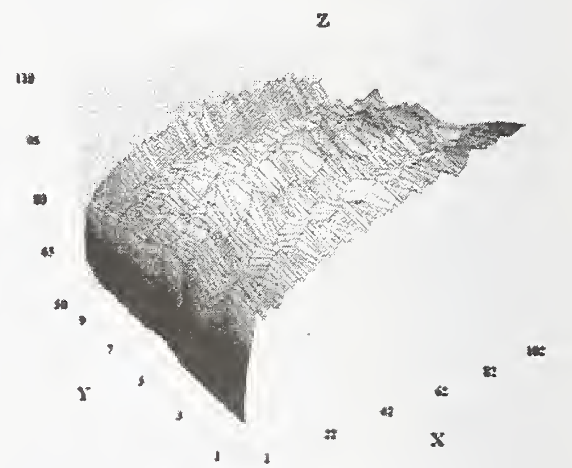


Figure 3 ASTRAL Generated Transmission Loss Surface

Figure 4 shows the transmission loss surface formed by the interpolation values at the same grid points as in Figure 2. The interpolator learned the features from the first half of the range data, within 50 nmi. Figures 3 and 4 show the matching surface on the left half of the surfaces (shorter ranges < 50 nmi), while missing some characteristics on the right half of the surfaces (beyond 50 nmi). When we included all of the range data to train the interpolator, the spike on the top-left end of the surface in Figure 4 was clearly shown on the surface formed by the interpolation values resulting from that interpolator. The RMS error for the transmission loss surface in Figure 4 is 0.5 dB, about 1 to 2% of the actual loss values. The interpolator surface showed in Figure 4 takes 360 iterations, for real-time operation 200 iterations would be sufficient for a 25-nmi range operation area; the RMS error for the smaller range operation area was about 1 dB in the case mentioned in the abstract. Using the more detailed simplification models, $\hat{f}$ and $\hat{g}$, would have less RMS errors, but they will take a longer time to train.

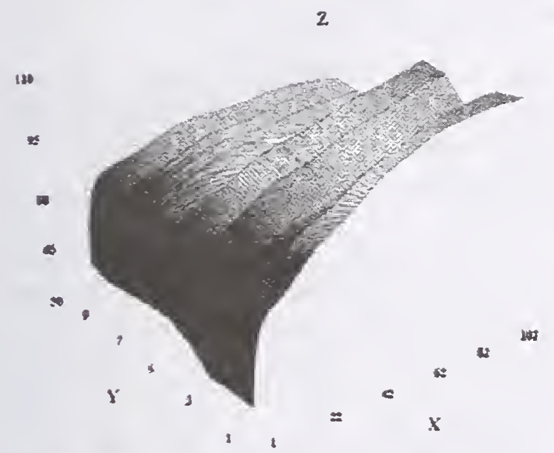


Figure 4 Interpolated and Extrapolated Transmission Loss Surface.

6.S ummary

This paper presents a model-fitting technique for use as an interpolator for underwater acoustic transmission loss. This interpolator could be useful in creating function approximation for one local domain in the adaptive interpolator discussed in Spall [8]. The error of this interpolator was tolerable and the computation speed was adequate for real-time training operations. There is a tradeoff between accuracy and desired speed — the more accuracy required the more time required for the NN to train. There is also a tradeoff between the accuracy and the complexity of the operational area. There is room for improvement in the optimum source-receiver pair selection and in the estimation model formulations. This technique could be used for extrapolation and inversion processes.

Acknowledgements

This work was supported by DARPA under contract MDA972-96-D-0002 in support of the Advanced Simulation Technology Thrust Area, Multiresolution Interaction Validity Project.

References

[1] Chin, D. C., "Simultaneous Perturbation Method for Processing Magnetospheric Images," 1999, *Optical Engineering*, v. 4, pp. 606-611.

[2] Spall, J. C., "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," 1992, *IEEE Trans. Autom. Control*, v. 37, pp. 332-341.

[3] Ng, G. W., Editor, *Application of Neural Networks to Adaptive Control of Nonlinear Systems*, 1997, Research Studies Press. LTD. & John Wiley & Son Inc., ISBN 0471 97263 0.

[4] Gohberg, I., Editor, *Time-Variant Systems and Interpolation*, 1992, Birkchasuer Verleg Basel, ISBN 0-8176-2738-3.

[5] "FY98 Progress Report; Multiresolution Interaction Validity Project," 1998, *Advanced Simulation Technology Trust*, STD-R-2774.

[6] *Software Test Description for the ASTRAL Model*, Naval Oceanographic Office, NRL OAML-STD-23E, 1994.

[7] Urick, R. J., *Principles of Underwater Sound*, 1976, McGraw-Hill, ISBN 0-932146-62-7.

[8] Spall, J. C., "Adaptive Model Fitting with Time-Varying Input Variables," 1999, *The Proceedings of 1999 American Control Conference*, San Diego, CA, pp. 1435 - 1440.

A Three-Tier Communication and Control Structure for the Distributed Simulation of an Automated Highway System *

R. Maarfi, E. L. Brown and S. Ramaswamy

Software Automation and Intelligence Laboratory, Department of Computer Science
Tennessee Technological University, Cookeville, TN 38505
Phone: (931)-372-3691. Email: srini@acm.org / srini@ieee.org

ABSTRACT: - This paper presents an Automated Highway Simulation using a hierarchy of communication to resolve advanced traffic situations. The simulation is being done to study communication paradigms, traffic concepts, and road design. The simulation is three dimensional; projected in a two-dimensional space, and implemented as a client/server and peer-to-peer system using Java. Vehicles communicate with each other using a linked graph called a 6 way DSM, or dynamic socket mesh. Cars also use a higher communication layer - called a segment controller, and segment controllers coordinate with each other using the simulation controller.

I. INTRODUCTION

This paper presents the simulation of an automated highway system with intelligent vehicles using higher-level, off-road intelligence to aid in their guidance. This simulation is being designed to study communication paradigms, road design, and self-correcting systems. The concept of an automated highway system, or AHS, is nothing new. An AHS is a system that allows vehicles to drive themselves without human intervention. General Motors Corporation first introduced AHS in the 1939 World Fair. This system, and others that have followed, have made necessary the complete absence of human drivers. This simulation does not have that limitation. The main objectives of an automated highway system are not unlike any other system. The system should do its job as efficiently and as safely as possible while trying to take all errors that can occur into account. The system is designed to work around errors that may otherwise prevent it from functioning.

It is important to mention that a few key concepts have emanated from a four-layer protocol used in Berkley's PATH laboratories, funded by the Federal Highway Authority. The three-tier communication system discussed in this paper is designed to anticipate errors that may occur inside any part of the communication system itself, and will eventually be aware of mechanical failures that can occur and give the vehicle or the driver instructions. If any part of the communication system should fail, the remaining parts will change to a different operating mode to compensate for the failure. Each layer of the three-tier system is necessary for full functionality, but any layer can be removed during the simulation without catastrophic effects.

Platooning is the most obvious method available to increase road traffic density. A platoon is a group of vehicles with just enough space between them to react to each other, possibly even moving at predefined speeds. Previous systems, however, have relied on slightly slower speeds and closer vehicle proximity for a highway utilization increase of four times what human drivers use today. This increases the capacity of the highway by properly utilizing available space. Platoons are isolated from each other, and from human drivers, by large gaps. Earlier studies have shown that platoons can be created without having an adverse effect on safety. In this simulation, since we allow human-controlled cars, platoons are restricted. This control is a parameter of the simulation controlled by the second tier. Cars will then stay at a two-second following distance and switch lanes to maneuver around vehicles when necessary. With other humans around, the cars can be made to follow the normal rules of the road.

An AHS is one of the most appropriate applications to test a design for a self-correcting system because of the unpredictability factors, specifically humans and mechanical failure. Since human-controlled cars are allowed in the system, there is even greater unpredictability. Weather can seriously alter the performance of a vehicle also, but this is not currently added to the simulation. Communication layer failure is already implemented.

The simulation is implemented as a client/server networking application, written in Java. Vehicle to vehicle communication is allowed via peer-to-peer network connections, and communication between the segment controllers, the simulation controller and the vehicles are handled by client-server methods. This networking environment allows for a suitable way to test communication protocol and efficiency. The system was designed for efficiency in terms of processor utilization and network utilization. Each layer is handled at the lowest level possible, thus minimizing the need for passing information around the network. This is illustrated in Figure 1. The communication architecture is divided into three layers:

i. Vehicle communication – Possibly representing sensors or radio communication, this is a way for cars to get information about the immediate surrounding cars such as speed and position. In the simulation this is implemented by means of a linked list structure, the dynamic socket mesh or DSM.

ii. Segment Controller – Segment controllers are local to specific sections of the highway. These sections are predefined internally, or are given to each segment controller by the simulation controller. Each vehicle only needs to communicate with the segment controller unless it is unavailable, then reaching the simulation controller is necessary. The segment controller informs a car of a malfunction in the six-way DSM.

iii. Simulation Controller – Responsible for assigning sections of the highway to segment controllers. The simulation controller may also act as a segment controller if a segment controller is unavailable. Future plans involve routing of traffic during construction or accidents.

The paper is organized into four sections. In section II, the vehicle communication, segment controller and simulation controller are described along with their respective issues. In section III, the details of implementation are discussed. Section IV concludes the paper.

II. DESIGN ISSUES

In our research, we have noticed that some limited intelligence need to be available to the vehicles themselves. Intelligent vehicles are a solution to the overhead normally associated with previous roadside-control design bottlenecks. In order for vehicles to behave intelligently, they must first be able to communicate with each other and share some required information. This section describes the design architecture for that communication and the application developed as a test-bed for evaluating performance. We assume three kinds of cars to be available in the simulation; viz. *(i)* Computer-Controlled Cars: These cars are the "smart" cars that can be controlled by the simulation and demonstrate the highest degree of intelligence. *(ii)* Human-Controlled Cars: These cars are controlled by human operators from a terminal and are completely unpredictable – similar to any highway scenario. *(iii)* Traffic Cars: These are "dumb" cars on the highway that just cruise along to create traffic situations – they are not controlled by any intelligent decision making process.

For our work the following three communication paradigms are readily applicable. The first, autonomous resolution; relies on receiving immediate data from the surrounding vehicles. This is a very limiting paradigm because in real-world situations vehicles may be unable to successfully request surrounding vehicles to even change lanes. The second approach, using a master/slave resolution, is not used unless a car is in a platoon. The platoon leader is then assumed to be the master, and directs computer-controlled cars in its platoon similar to the way a traffic cop directs traffic.

The third paradigm, the mutual resolution paradigm, allows vehicles in close proximity to switch lanes without running into each other and to allow cars to switch into the desired lane in heavy traffic. This paradigm is used in our simulation for resolving the majority of highway maneuvers, however, application of a particular paradigm depends on the environment in the simulation. For example, if a human controlled car is in front of a computer-controlled car, the human controlled car cannot be asked to move out of the way or change speeds. A computer-controlled car is also unable to switch lanes, unless it is able to get clearance, which again may not be possible with a human-controlled car in the way. Mutual resolution, however, would allow a computer-controlled car to switch lanes first if, for example, it needed to reach the exit that would otherwise be blocked. A central controller is not necessary for this situation if each car is able to communicate its current goals to the other car, and each car is aware of what to do depending on the adjacent cars' (shared information, if available) goals, its goals, and its surroundings. This is similar to a finite state machine. Each car in a normal situation makes its decisions based on a predetermined coordinating logic.

There are advanced traffic situations involving nonstandard vehicles, such as ambulances, roadblocks and closed lanes. Using sensor data (simulated data this implementation), a car may be able to see a blocked lane or a stopped car, but the delay in seeing the block and being able to switch lanes may cause significant traffic blockage.

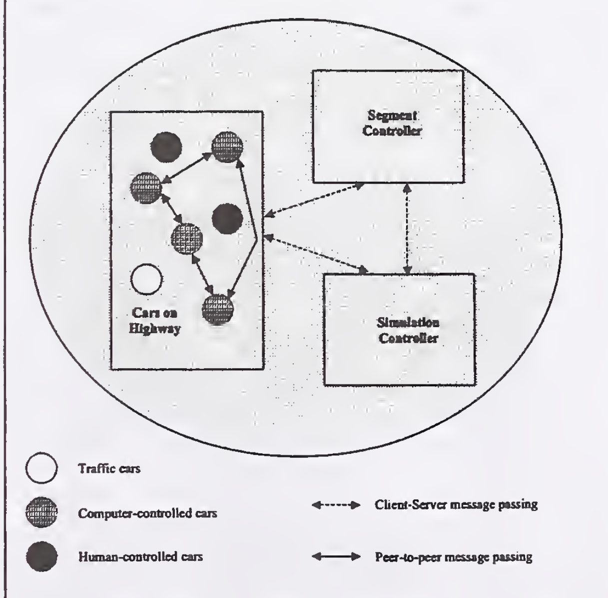


Figure 1. Control and Communication Overview

The same case applies with an ambulance, as shown in Figure 9 and Figure 10. Sensor data will allow a car to identify a vehicle and its speed, but it would be unable to switch lanes if another vehicle were to be in the way. While the implementation is currently not present, a car might also be advised to take a detour if an accident has occurred or road construction prevents passage. These and similar tasks are handled by the segment controller. Each segment controller is responsible for an arbitrarily large section of the highway. The segment controller also informs cars of the existence of human drivers for resolving the decision to platoon and the speed at which to travel.

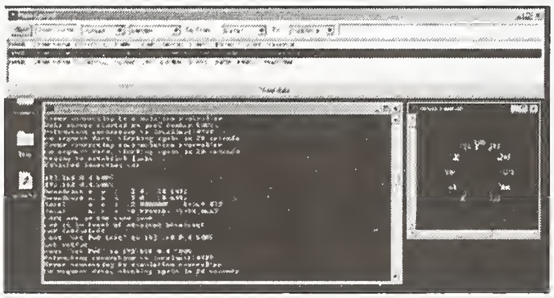


Figure 2. Four types of cars in simulation

Since a segment controller only oversees a relatively small, portion of the AHS, a higher communication level is necessary for the controllers to be aware of unfavorable occurrences down the road, such as weather, traffic, or road construction, between segments and to assign segments of the highway to their respective controllers. This task is given to the simulation controller. Without the simulation controller, vehicles are unaware of intelligent decision making to decide upon the most efficient route to their destination. Besides, the simulation controller also is in-charge of initially setting up the necessary parameters of the simulation.

Each level of communication, however, is designed to function in the absence of the other. For example, if a car's sensor should fail, the segment controller will be aware of the discrepancy between what one vehicle is reporting and what another vehicle is reporting. Once aware, the segment controller is capable of guiding a car to the nearest exit by relaying information that vehicles are reporting through the segment controller or, depending on the existence of humans in the segment, merely instruct the car to pull over or for the human driver to assume control. If a segment controller should fail, the simulation controller has the ability to function as a temporary segment controller. The vehicles attempt communication with the simulation controller automatically whenever failure with the segment controller occurs. This also allows for segment controllers to be changed without affecting the simulation. If both the simulation and segment controllers should fail, the vehicles will still be capable of navigating the highway, but they will be unable to platoon since the existence of human drivers is unknown. They will be unaware of advanced traffic situations.

As shown above in Figure 2, three control types are available. Traffic 1 represents a car that is unaware of its environment and is only there to be an obstacle to other vehicles. Traffic 1 never makes a lane change, however, Traffic 2 randomly changes lanes. It will not change lanes into other cars, but provides an advanced challenge to the computer-controlled cars. The computer-controlled car is capable of joining or leading platoons, and it is capable of intelligently navigating around traffic and human-controlled cars. Human-controlled cars are completely driven by an operator at the terminal. The car controls are shown in the mid-right section and are capable of allowing lane changes or speed changes. The speedometer shows the current speed, and the bottom scrollbar shows the current lane.

III. IMPLEMENTATION

Figure 3. Links and Visualization

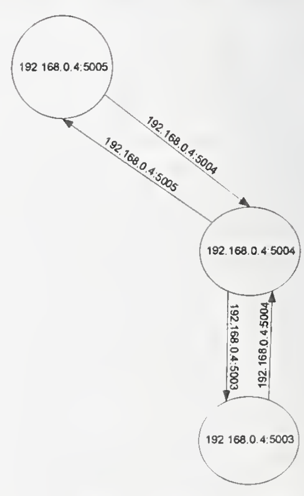


Figure 4. Internal representation of links

Effective networking, avoiding delays and redundancy, is critical to this project because of the representation of vehicle communication between each other, between segment controllers and between simulation controllers. To simulate real-world radio/wireless communications, the simulation uses TCP/IP sockets to communicate with each other. Necessary sensors are also represented as sockets in the implementation.

Inter-vehicle communication is represented as a series of six sockets, called a six-way dynamic socket mesh, or DSM. Each direction represents what the vehicle would see at a given location, specifically forward, forward-right, backward, backward-right, backward-left, and forward-left. This is a linked structure that forms a graph, and each car is a point on the graph. This represents sensory information gathering in a real-world implementation, so they are able to get basic operational data from these connections. However, using the mutual resolution paradigm allows for cars to communicate with each other to exchange awareness of their current goals – this will need more than just sensing capabilities; i.e. intelligent message passing, if implemented in a real-world scenario. However, to mimic expected reality, this is assumed to only happen between two computer-controlled cars.

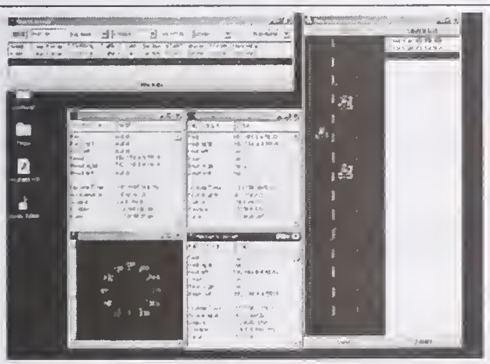


Figure 5. Passing the first green car

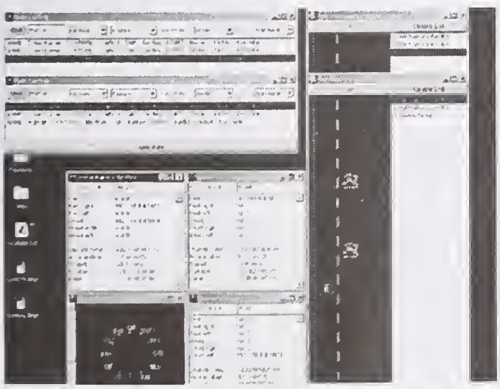


Figure 6. Passing the second green car

Automatic testing programs were written to ensure proper functioning of the simulation. In addition to testing programs, a rudimentary visualization was implemented to analyze the simulation. Java was the language of choice for the implementation, since it aids in portability and simplifies the distributed networking implementation. Figure 3 and Figure 4 demonstrate how the links are represented. The right hand side of Figure 3 shows the visualization application.

The dialog boxes shown in Figure 3 surrounding the speedometer represent the data stored by each car, which can be retrieved by double clicking on either the individual listing of the car in the main control application or by double clicking the list in the Visualization. This shows that each car identifies and locates others by storing IP addresses and sockets. Therefore, in the simulation, computer-controlled cars have the same link maintenance logic as human and traffic. The green cars in Figure 3, (and Figure 5, Figure 6) are traffic cars while the red car is a human-controlled car. It is to be noted that in the visualization, cars are traveling downward, so in this case the forward link is actually down, right is left, vice versa. In Figure 4, the circles represent cars and their unique nodes on the network, and the arrows pointing to other cars represent the connections on the graph. This concludes that a link is represented by two TCP/IP socket connections, which is similar to sensors mounted on the two cars.

In Figure 5 and Figure 6, the continuing validity of the links is illustrated. It shows that as the red car passes the first green car, it looses its forward right connection to it and gains a backward right. Conversely, the green car gains a forward left and looses a backward left. The front green car gains a backward left. Note that the backward left link was invalid to have by definition of the graph until the first green car had been passed.

As mentioned earlier, the segment controller is in charge of all vehicles within a small portion of the highway. When a segment controller is launched, it connects to the

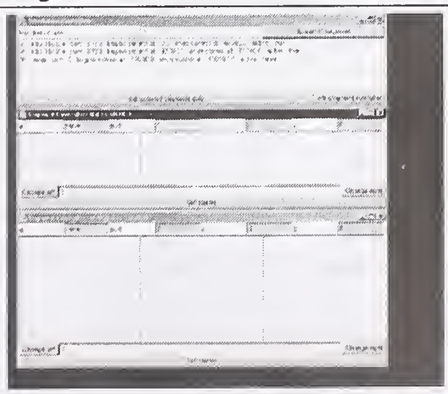


Figure 7. Simulation and segment controllers

simulation controller to determine what area of influence it has. If the simulation controller is unavailable, it reads from a file what section it is supposed to be controlling. This allows functionality without the simulation controller, but allows the simulation controller to assign new areas if change is necessary. Unlike the segment controller, each vehicle first attempts to connect its last known segment controller for the area it is currently located. If no segment is available, the car then attempts to connect to the simulation controller to locate another segment controller. If the simulation controller does not have a predefined segment for the vehicles location, the car is informed. If a segment is predefined but a controller does not exist, the simulation controller launches a segment controller for it to connect to. This means that the simulation controller actually runs segment controller logic if a segment is absent, and tells the car to connect to it.

Figure 7 shows two segment controllers executing and the simulation controller at the top of the screen. The three lines (of text) at the top of the simulation controller represent a predefined segment. The first two are active, and the last is inactive. The three white areas in the middle of the segment controllers (no cars are located in either segment) represent each lane of the highway. The change right and change left buttons allow for the movement of cars

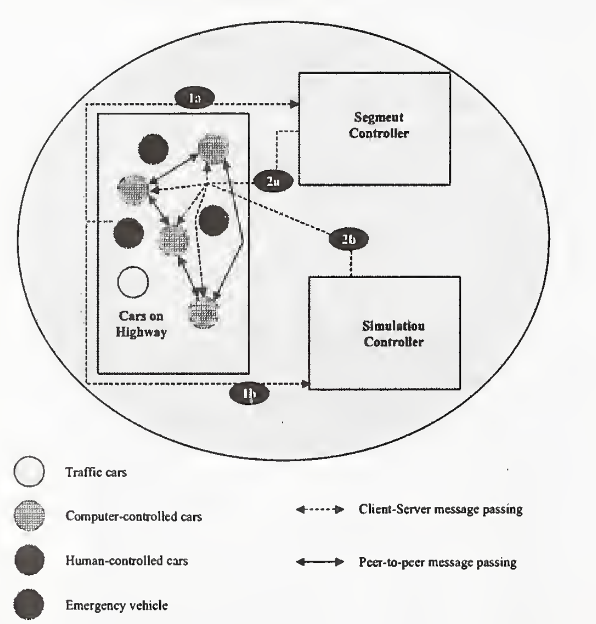


Figure 8. Communication Flow in an Emergency Situation

by the segment controller for testing purposes, as does the set speed.

A useful application of the simulation controller can be seen in Figure 8. Figure 9 and Figure 10 show the corresponding implementation for the scenario depicted in Figure 8. Here, a computer-controlled vehicle – an ambulance, gets priority in using a lane and all computer-controlled cars are instructed to move to a different lane. The ambulance will use either the segment controller or the simulation controller to clear its lane. In the case illustrated in Figure 9, an ambulance vehicle has just been informed of an emergency, but two cars (a red human controlled car – immediately upfront and a green computer-controlled car before the human controlled car) are in the way. The human-controlled car (line of sight communication) may not immediately move over while the other car is a computer-controlled car. Since the ambulance is unable to directly see

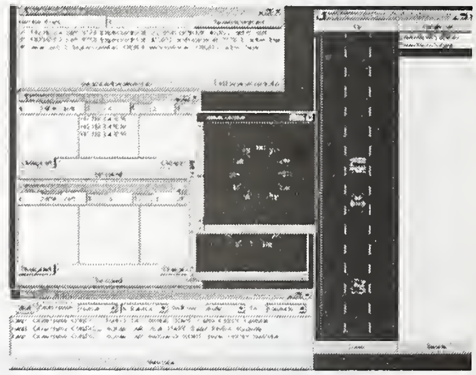


Figure 9. Ambulance wishes to pass

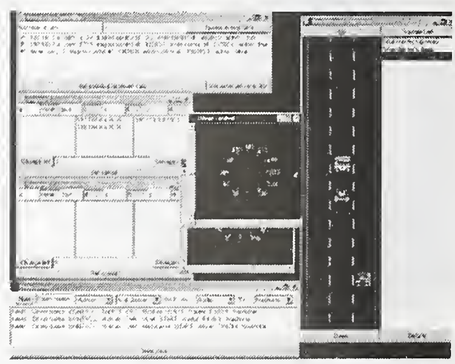


Figure 10. Ambulance can only move computer cars out of its way

the computer-controlled car, it would be unable to inform the car to move out of the way. However, with the segment controller, the ambulance sends a message to the segment controller, which, in turn, sends a message to every computer or traffic-controlled car in the lane, within a predefined range, commanding it to move out of the way. Figure 10 shows the segment controller's ability to request the green car (computer-controlled) out of the ambulance's way, even though the human car (one right before the ambulance in the middle lane) is blocking its view.

The segment controller increases network efficiency in cases with more cars because each node on the graph does not have to be traversed to relay messages. However, a computer-controlled car would only be able to dodge the ambulance if it was directly in front of it.

More importantly, a lane lock has occurred in the middle lane. Set by the segment controller, computer controlled cars are unable to switch back into the middle lane until the ambulance is out of the way. To increase network efficiency, vehicles are unaware of the lane lock until attempting to switch into the lane. This allows cars to switch lanes without informing the segment controller. Without the segment controller, cars would have to check their diagonal links for situations like an ambulance, but due to the functionality of the links demonstrated in Figure 3, Figure 5, and Figure 6, they might still not have seen the ambulance coming until it began to switch lanes.

IV. CONCLUSION

This paper presented the implementation of a flexible and robust three-tier communication and control structure applied to a distributed simulation of an Automated Highway Simulation. Vehicles can be controlled by humans or be logic driven. This paper captures the most likely scenario of any future AHS as a result. The Java platform provides adaptability and object-oriented structure that allows for modularization for easier future modification and maintenance, and allows portability. The paper has also shown that while external communication is not necessary for functionality, efficiency is greatly increased and safety in the face of failure is ensured.

V. REFERENCES

[1] R. E. Fenton and R. J. Mayhan, "Automated Highway Studies at Ohio State University- An Overview," *IEEE Trans. on Vehicular Technology*, vol. 40, no. 1, pp. 100-113, Feb. 1991.

[2] C. Unsal, P. Kachroo and J. S. Bay, "Multiple Stochastic Learning Automata for Vehicle Path Control in an Automated Highway System," *IEEE Trans on Systems, Man and Cybernetics*, vol. 29, no.1, pp. 120-127, Jan. 1999.

[3] D.N. Godbole, J. Lygeros, E. Singh, A. Deshpande, A.E. Lindsey. "Communication Protocols for a Fault Tolerant Automated Highway System". Dept of Elec. Engg. and Comp. Sciences. Univ. of CA.

[4] R. Donath, April Dean, D. Girardi and S. Ramaswamy, " Distributed Simulation of an Automated Highway System with Intelligent Vehicles ", *2001 Summer Computer Simulation Conf*, July 2001, Orlando, Florida.

[5] Luke D. Postema, "Automated Highway Systems: Incremental Deployment as a Solution for The Future of Transportation". Washington internships for Students of Engineering. August 1998

Tessellating and Searching Uncertain State Spaces

A. Meystel, A. Bathija

Department of Electrical and Computer Engineering, Drexel University
Philadelphia, PA 19104

Abstract

Multiresolutional S^3-search generated a need to properly tessellate spaces and efficiently searching them. Drexel University has introduced MR-methodology such as uniform and non-uniform space tessellation and efficient algorithms of searching within ressellated state space. This methodology for solving planning and control problems is successfully applied in autonomous vehicles, industrial robots and power stations. This paper focuses on computational phenomena characteristic for randomized tessellation and affecting the results of S^3-search.

Keywords: bias, Dijkstra algorithm, envelope, multiresolutional, randomized tessellation, shaking the grid, search in the state space, stripe, uniform and non-uniform tessellation, variable traversability

Introduction

To "tessellate" to form or arrange elementary units of space in a mosaic fashion so that all selected space be covered. Tessellation of a closed polytope means decomposing it into non-overlapping sets of tessellata (granules, boxes or tiles). These subsets form an equivalence class in which all the points within a tile are identified with a single label. The natural interpretation is that each of the labels represents all points of a single tessellatum while its generalized location is in the center of the tessellatum. There are three basic types of tessellations:

1. Regular Tessellation
2. Semi Regular Tessellation
3. Irregular Tessellation

Regular Tessellation: This means a tessellation made up of congruent regular polygons. *Regular* means that the sides of the polygon are all the same length. *Congruent* means that the polygons that you put together are all the same size and shape. Only three regular polygons tessellate in the Euclidean plane: triangles, squares or hexagons. Regular tessellation are shown in figure 1.a.

Semi-Regular Tessellation: A variety of regular polygons is used to make **semi-regular tessellations**. Examples of this type of tessellation are shown in Figure 1.b. This tessellation has two properties:

1. It is formed by regular polygons.
2. The arrangement of polygons at every vertex point is identical.

Irregular Tessellation: In this case points are sprinkled randomly. This randomity is equivalent to the uncertainties o sampling the state space. There are quite a few methods of generating random tessellations.

Generating randomized tessellation.

S^3-search requires generating random points in the state space [1]. The law of distribution that characterizes coordinates of the random points describes the uncertainty of the state space taken into consideration. There are

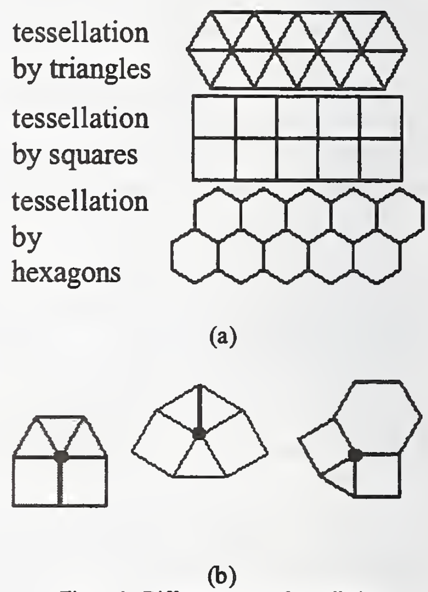


Figure 1: Different types of tessellation

many methods of generating randomized graphs (Irregular tessellation). A few of them are listed below.

1. ***Random generation of grid-points positions*:** Random points in the state space can be generated by a standard random number generator. A deficiency of this method is in the fact that "random" points may not be evenly distributed. In other words, the density of the particular set of points in all regions of the state

space may not be the same. This may occur because of dealing with imperfect random number generators. However, this shortcoming can be dealt with by multiple running the algorithm that is being tested.

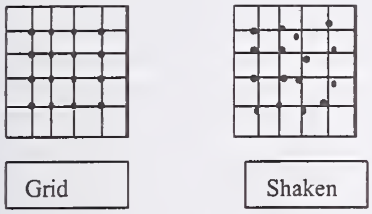


Figure 2: Shaken grid method of generating random points

2. ***Randomizing by "shaking the grid":*** In this method we use a grid to generate the basic set of points. The grid points are considered to be at the intersection of a row and a column. Then we introduce a random shift to these coordinates of points: shake the grid. Thus, the points move away from the initial positions into random positions.

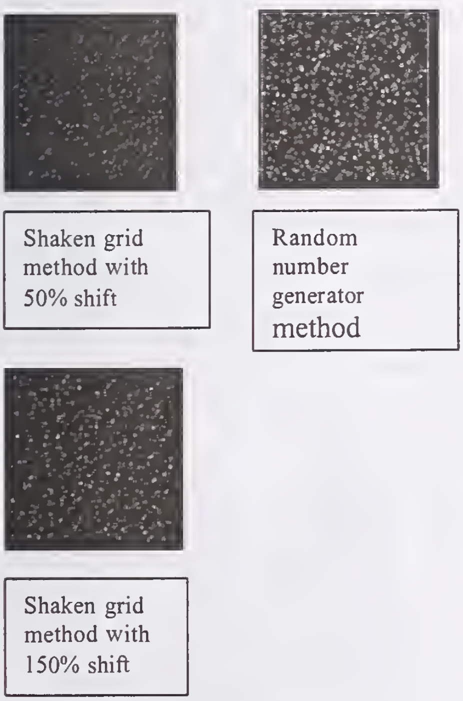


Figure 3: Comparison of the various methods for generating the grid

The shift is assigned in the form of a fraction (percentage) of the interval between the rows and the columns (Δ). Figure 2 explains the generation of the randomized tessellation using the grid method. The advantage of this method is that it ensures that the *overall density* of the points is the same in all the regions of the graph. But this true only if the shift in the points is less than or equal to 50% (.5Δ). If the shift exceeds 50% the distribution gets similar to the one observed in the random number generation method. This comparison is shown in Figure 3.

3. ***Concentric circle non-uniform grid generation:*** This method is similar to the *grid generation method.* In the latter we used straight lines to generate the grid. But in this case we use concentric circles instead of the rows and the radius replaces columns. The specifics of this method is that there are many points generated close to the center of the circle and as we move away from the center the density of the points decrease. This method is useful in the cases when a variable resolution is attempted within one level of representation like in page 4 of [2], where the fine motion planning is obtained for a few steps in the vicinity and the further we move from the acting robot the lower the resolution is. Certainly, any law of gradual decrease of resolution can be assigned. In Figure 4: we illustrate how the random points are generated by this method.

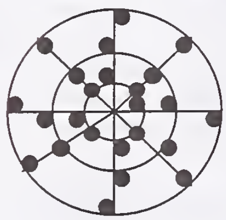


Figure 4: Non-uniform Random points generation using concentric circles

Randomized Grid Generation

The algorithm of randomized grid generation is described by this pseudocode:

1. Start.
2. Obtain the parameter of the grid
   a. Total number of points
   b. Total number of rows
   c. Total number of columns
   d. Check if the product of the rows and the columns is equal to the total number of points. It true then continue else go to step 2.a
   e. Obtain the starting coordinate of the row and the column.
   f. Obtain the delta for the row. Check if the obtained value is valid.

g. Obtain the delta for the column. Check if the entered value is valid
h. Obtain the shift percentage of the points.
i. Calculate the max possible shift of a point with respect to the row and the column delta.

3. Repeat the following for total number of rows (columns).
    i. Assign a random sign (+ or -)
    ii. Obtain a random value with the max possible shifts for each of the row and column.
    iii. Add this shift to the corresponding row and column coordinate.
4. Store this information
5. Stop.

Randomized graph using different distributions.

In the above pseudocode, we assign the "sign" that a random generator uses to obtain the random shift within the maximum limits. Different distributions can be used to obtain this random value. Uniform distribution is the distribution that we have used in our experiments. In uniform distribution the random point obtained has equal probability of falling anywhere in the space specified for it. While if we use the Guassian Distribution then the points have a greater probability of falling close to the center of the shift compared to the probability of falling at the edge of the maximum shifts.

Searching on the Grid Problem: Why Bias?

This is a useful advantage and the disadvantage of a grid that from any node in the grid to another there are many equal (or approximately equal) paths. This can be seen in Figure 5.

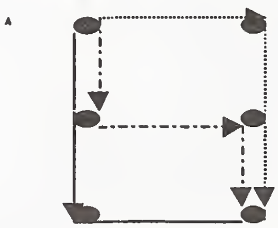


Figure 5: Three possible paths from node A to node B in the grid

In the above Figure, it is clear that there are 3 possible equivalent paths from node A to node B. On using the new algorithm on the grid, it tends to give the full line path (in Figure 5) as the optimal solution. It turns out to be that if the starting node is on the left hand side and the destination point is on the right hand side and there are many possible paths from the starting to the destination then the left most path will be given as the optimal solution. Similarly if the starting point in the right hand side and the destination on the left hand side, then the right most path among the possible paths will be given as the optimal one. For a single run of the min-cost search see Figure 6.

The concept of Vicinity

Unlike the standard Dijkstra search algorithm, our search in the randomized grid does not have any graph prepared for exploration in advance for exploration. The recommended algorithms builds the graph as it explores the graph. The concept of a vicinity is introduced. The present "standpoint" is being surrounded by a "vicinity" that can have a "radius" of 1, 2, (or more) of average edge length (v1, v2, v3). Edges are constructed to all nodes in the vicinity. After the current "cheapest" node is found, it becomes the standpoint node and is being surrounded by its vicinity. Certainly, the solutions of 3-vicinity lead to "straighter" trajectories, but it takes more complexity to compute them.

Superimposing Multiple Search Results

A better way of finding the generalized result of searching in the randomized graph is running search in multiple random grids and superimposing their solution. In Figure 6, we show the results of such superimposing of several min-cost paths for a dynamic system of

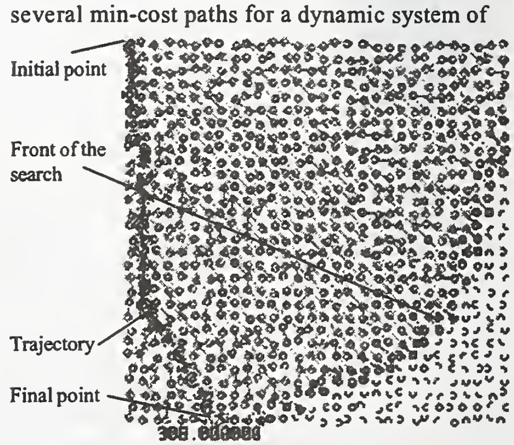


Figure 6: Velocity vs Distance with Friction (v1)

the first order (the acceleration and deceleration of a constant mass on a surface with friction. One can see that acceleration is not as rapid as deceleration. (This test further develops our prior results from [3]). It is important that in a multiplicity of superimposed random search results all stochastic components compensate for each other, and the envelope (the stripe for the subsequent higher resolution search) has smooth edges (as seen in Figure 9).

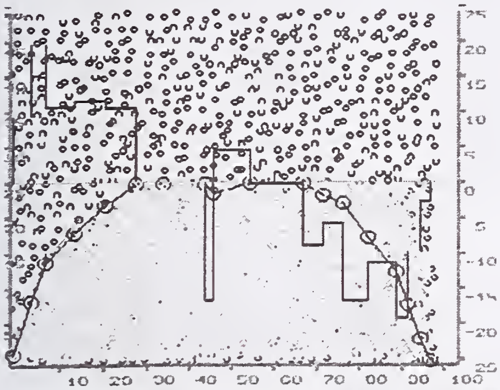


Figure 7: A single search performed on a grid with a shift of 30% and vicinity 1.

Solution to the grid problem

To avoid getting the extreme paths as the solutions we have to make sure that there aren't any "equidistant paths" from the starting to the destination point. This can be achieved by increasing the shift in the points there by eliminating the distorting effect of the grid.

(a) (b)

Figure 8. (a) Super imposed image of 30 runs of 30% shift (b) Super imposed image of 30 runs of 60% shift. The above runs were for performing a search of optimal path from the (0,0) location in the grid to (29,7) location

A comparison of the search with shift of 30% and 60% is shown in Figure 8 as the results of searching optimal path from the (0,0) location in the grid to (29,7) location It is clear from the Figure 8, the phenomenon of Bias is due to the grid effect. It was found that 50% shift is the most beneficial for removing the distorting grid effect. In other words, by shifting from a regular tessellation to a irregular tessellation we can reduce the bias problem substantially. Large bias is shown in Figure 6, a reduced case in Figure 8.

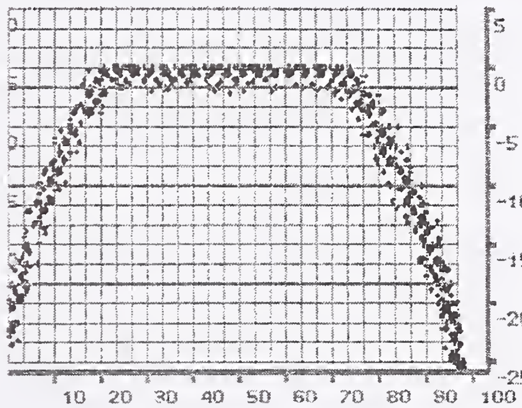


Figure 9. Envelope ("stripe") for the subsequent search of higher resolution obtained by superimposing multiple results of randomized searching.

Using randomization of the grid for virtual modeling the uncertainties

While using the tessellation for any particular purpose, 3 components affect actively the results of the execution.

1. *Grid Law:* The law is determined by the technique that we use to generate the tessellation.
2. *Grid Density:* This is computed as a ratio: the quantity of points located in the state space over its volume (area).
3. *Shift of the randomization:* This is the Results of "shaking" the grid (percentage) introduced to generate the effect of randomization.

Among these 3 components, the grid density and the shift of randomization create an intrinsic error of the path. When solving the same problem analytically, the solution (path) obtained has a stochastic component in it, which is perceived as a noise, a part of the uncertainties of the sources. This stochastic component is equivalent to the intrinsic error. Hence, if we know the sources of the uncertainty in the system (e. g. in the form of the values of Expectation and the Variance of this uncertainty) we can assign the grid density and the shift and the results of randomization will be equivalent to using the models of stochastic components, since we obtain the same Expectation and the Variance as the model would produce. In other words, instead of modeling the uncertainties we are can build the tessellation in such a manner that is produces this statistical truth.

Randomization in the Case of Multiple Traversabilities Segments

It is instructive to consider a case of finding the minimum cost path for a case of having multiple traversabilities of the state space. In Figure 10, a space segmented in multiple zones of traversbility is shown and the results of vl-searching for the case when traversability gradually reduces clockwise.

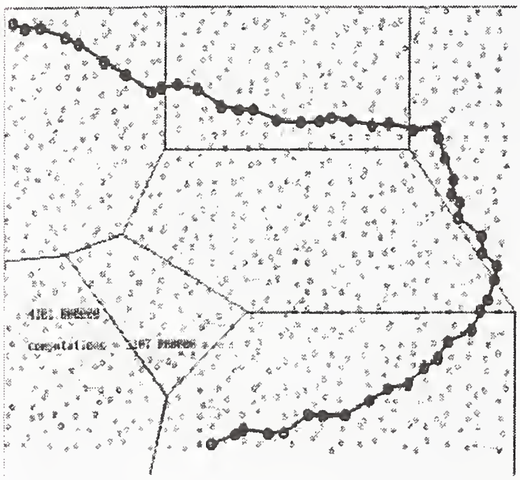


Figure 10. A single run of the S^3-search algorithm and a randomized grid

A question that should be answered: how to determine the width of a stripe (envelope) within which the search should be executed at a higher level of resolution. We exercised computing the boundaries for a stripe with 3σ width on the right and on the left around the average approximation of the path trajectory. This is a pretty cumbersome computation that includes the following stages: a) finding the approximation of the single path known, b) assuming that this approximation can be considered "an average recommended trajectory," c) computing the value of 3σ from the information of uncertainty, d) constructing piecewise boundaries around the average trajectory.

The following technique was tested and seems to be more practical. The search is being run many times, and the results of this multiple search are collected together and are considered a "stripe" (envelope).

In Figure 11, the results of this experiment are brought together for the case shown in Figure 10.

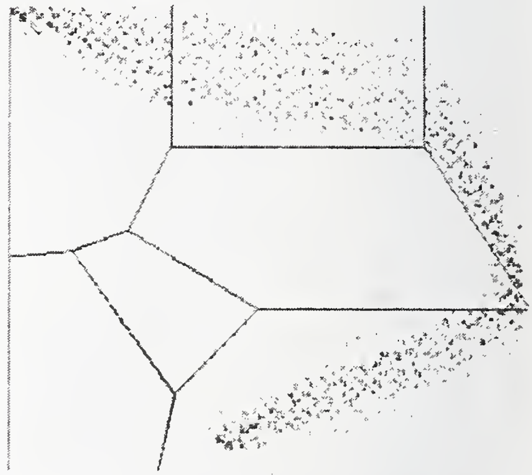


Figure 11. The cumulative results of conducting multiple search for thecase shown in Figure 10.

One can see that a stripe emerges that demonstrate the statistical representation of the zone that is preferential for the subsequent higher resolution searching.

Interesting observations can be made: a) the stripe is narrowing as the traversability of tthe path is reducing, b) the stripe is narrowing in the areas of leaving IP and reaching the GP, c) in some areas the distribution gravitates to the unimodal law (like in Figure 8b); however in many areas the distribution is similar to the uniform one

References:

[1] A. Meystel, J. Albus. *Intelligent Systems: Architectures, Design, Control,* Wiley, NY, 2002, Pages 381-387

[2] A. Meystel, "Learning-Planning-Control Continuum," Ed. by H. S. Sarjoughian, et al, *2000 AI, Simulation and Planning in High Autonomy Systems,* SCS, San Diego, CA, 2000, pp. 3-23

3. A. Meystel, A. Lacaze, "Introduction to Integrated Planning-Learning Paradigm", in Proceedings of the 1997 International Conference on Intelligent Systems and Semiotics, Gaithersburg, MD, 1997, pp. 522-530

Autonomy and Socialization

Abstract

Kirstie L. Bellman, Ph.D.
AISC
The Aerospace Corporation

For quite some time it has been popular to study intelligence within the context of autonomous systems. When studying autonomous systems, the focus is often on the adaptive perceptual, behavioral and cognitive capabilities of an individual within some operational environment. However, if we look at the behavior of biological systems, we see dependency on other individuals as well as autonomy, and capabilities for coping as a member of a population as well as coping as an individual. Of course, there is enormous diversity in how frequently or in what manner the members of a given biological population interact.

In the past, many researchers, while they recognized the importance of animal populations and group behaviors, took the scientific strategy of focusing first on discovering the mechanisms for the individual's perception and behavior. They assumed that such behaviors were somehow more fundamental than group behavior. The strategy assumed that they could later tackle the additional capabilities needed for social, cooperative, or interactive behaviors. This paper presents the view that social behaviors are as fundamentally part of the construction of intelligent behavioral capabilities and as essential for survival as any individual's perceptual, cognitive or behavioral capabilities. This paper also presents our reasons for believing that some type of analogues of social capabilities are necessary to all autonomous constructed systems, such as "agents" or robots, if we want intelligent, independent behavior in real world environments. We will describe the types of capabilities necessary for adaptive behavior in both individual and social behavior. We will then discuss how social behaviors, cultures, and cooperative behaviors enhance the capabilities for individual autonomous behavior.

PART I
RESEARCH PAPERS

3P2 - Towards Universal Planning/Control Systems

1. A Sketch of Multiresolutional Decision Support Systems Theory
   A. Meystel, Drexel University

2. Motivated Metamodels
   P. K. Davis, RAND
   J. H. Bigelow, RAND

3. On the Role of Quality Attributes in Specifying Software/System Architecture for Intelligent Systems
   H. Sarjoughian, Arizona State University

4. Uncertain Predictions of Flow and Transport in Random Porous Media: The Implications for Process Planning and Control
   S. Orr, MRDS, Inc.

A Sketch of

Multiresolutional Decision Support Systems Theory

A. Meystel

Department of Electrical and Computer Engineering, Drexel University
Philadelphia, PA 19104

Abstract

Multiresolutional Decision Support Systems gain better performance and higher accuracy by the virtue of building highly efficient multiresolutional representation and employing multiscale Behavior Generation Subsystem (Planning and Control). The latter are equipped by devices for unsupervised learning that adjust their functioning to the results of self-identification. We show planning and learning to be joint processes.

Keyswords *behavior generation, control, decision support, generalization, knowledge, learning, instantiation, multiresolutional, multiscale, planning, randomized, representation, resolution, search*

1. Introduction

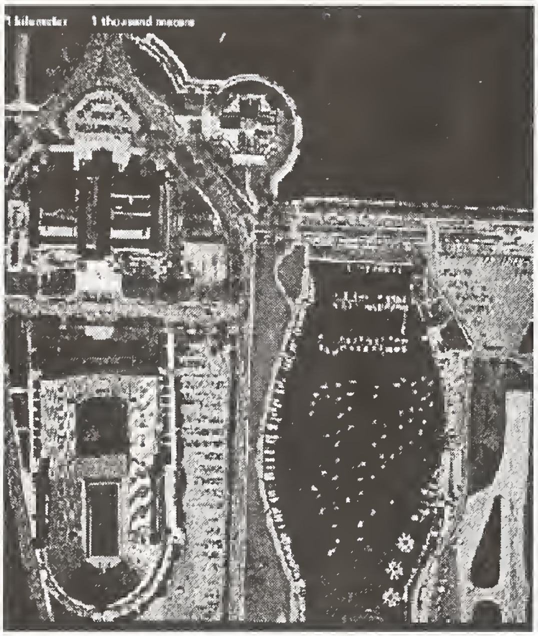
Multiresolutional Representation (MR) of the World should be considered one of the tools in the arsenal of Knowledge Management [1]. It is the tool that is widely applied but is scarcely noticed, probably because of its overwhelming omnipresence. The concept of MR can be illustrated by the series of pictures shown in Figure 1 (a - f). The enhanced set of these pictures with much more details can be seen in [2]. The resolution of each subsequent image is increased by an order of magnitude while the are of observation is simultaneously reduced by two orders of magnitude. It is not difficult to deduce that as far as the underlying knowledge is concerned, the objects in the image *f* are contained in the image *e* [(skin texture)⊃(hand)], the objects of the image *e* are contained in the image *d* [(hand)⊃(sleeping person)], the objects of the image *d* are contained in the image *c* [(sleeping person)⊃(picnic)], the objects of the image *c* are contained in the image *b* [(picnic)⊃(green lawn)], and the objects of the image *b* are contained in the image *a* [(green lawn)⊃(part of the city)].

This MR nestedness of sub-processes and sub-systems of the overall processes and system in not obvious in a standard cursory analysis, it can be discovered only as a result of special observations (computer vision equipment) and investigative analysis. More importantly, it is not obvious that this nestedness of entities and their properties is important (if necessary at all) for supporting the decision making activities at each level of resolution. Yet, all images in Figure 1 are tightly linked by the prior cognitive activities that are unified by identical processes of generalization performed upon higher resolution images to obtain a lower resolution image. Similar processes of instantiation allow for receiving each higher resolution image from the lower Resolution. Actually, not the process of sensing or the process of image edges detection and segmentation (and others) determine further image understanding and interpretation but rather the joint processes of generalization and instantiation that are executed upon these images top-down and bottom-up.

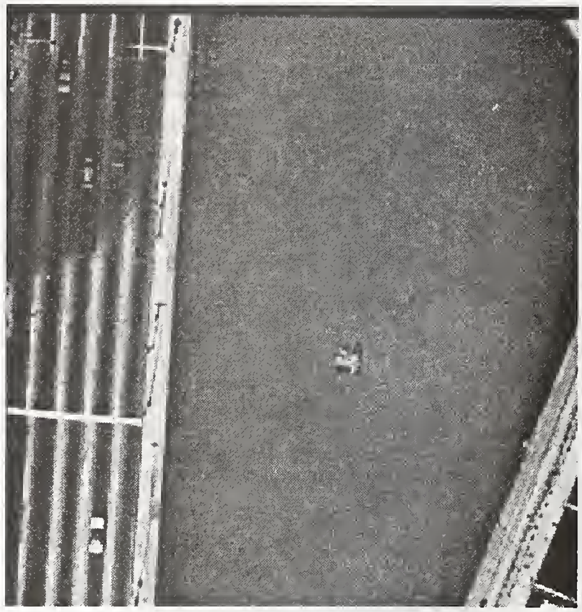
As the signals measuring and processing is conducted, at each particular level of resolution they contain a different package of frequencies (Figure 2). It demonstrates that the granularity of representation is correlated with the bandwidth of the signals at a level.

Why do we encounter this phenomenon: multiresolutional knowledge representation? Why the mechanisms emerged of generalization and instantiation? The reduction of complexity via reduction of "multiplicity" could only be done by the virtue of grouping and representing the group by a single symbol. This semiotic principle emerged because of the need to reduce computational burden. Computational benefits for a particular example of knowledge representation associated with planning is given in [8, 9].

The system of representation based upon recursive grouping/decomposition incorporates and uses the algorithms of generalization and instantiation in different incarnations that depend on circumstantial factors as for example, the information we are dealing with, or the subsystem of the world where the results are applied. Thus, the learning system must employ the same tools: labeling the entities in order to deal with concise notations (symbols), grouping the entities, decomposing them if information details are required. Learning systems use the same computational mechanisms.



a



b

e

f

Figure 1. Consecutive increasing the resolution of representation

The decision support system (DSS) treats knowledge as an entity suggested in [1]: it employs the awareness of familiarity gained by experience for storing experiences as well as for constructing decisions (including plans and controls) that ensure functioning of a goal-oriented system with increased performance index. MR gives an opportunity to minimize the value of computational complexity in a subset of DSS that organizes knowledge by joint processes of generalization and instantiation and use nested MR-search for converging to a recommended solution. This concept was introduced for planning and control purposes in 1986 [2] and explored in depth in subsequent works [3-10].

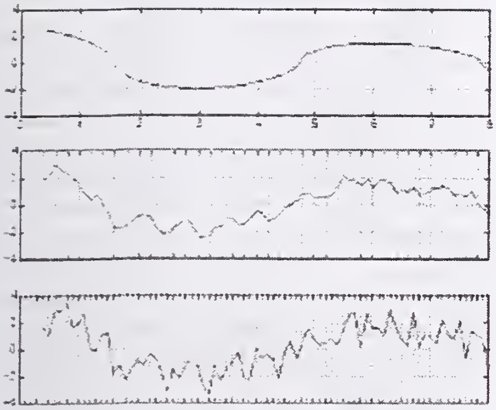


Figure 2. Multiresolutional representation (a-low resolution, b-mid-resolution+low resolution, c-the sum of signals of all levels)

All these processes dwell on the processes of learning employed by MR-DSS.

2. Decision Support of Behavior Generation

The structure of power station control system shown in Figure 3 was successfully tested at Delmarwa Power Station, DE, USA [11]. is required in all faculties of a system shown in Figure 3. Three levels of resolution are demonstrated Low ("Task Level"), Middle ("Component Level") and High ("Actuation Level"). Each level forms a loop closed through connection 1. Each of these loops is a loop of "closure" [9] and is equivalent to the Elementary Loop of Functioning (ELF) described in [6, 7, 8].

The vertical subsystem 5 (Plant) from Figure 3 is equivalent to subsystem Sensors, World and Actuators from the Elementary Loop of Functioning that is described in [8]..

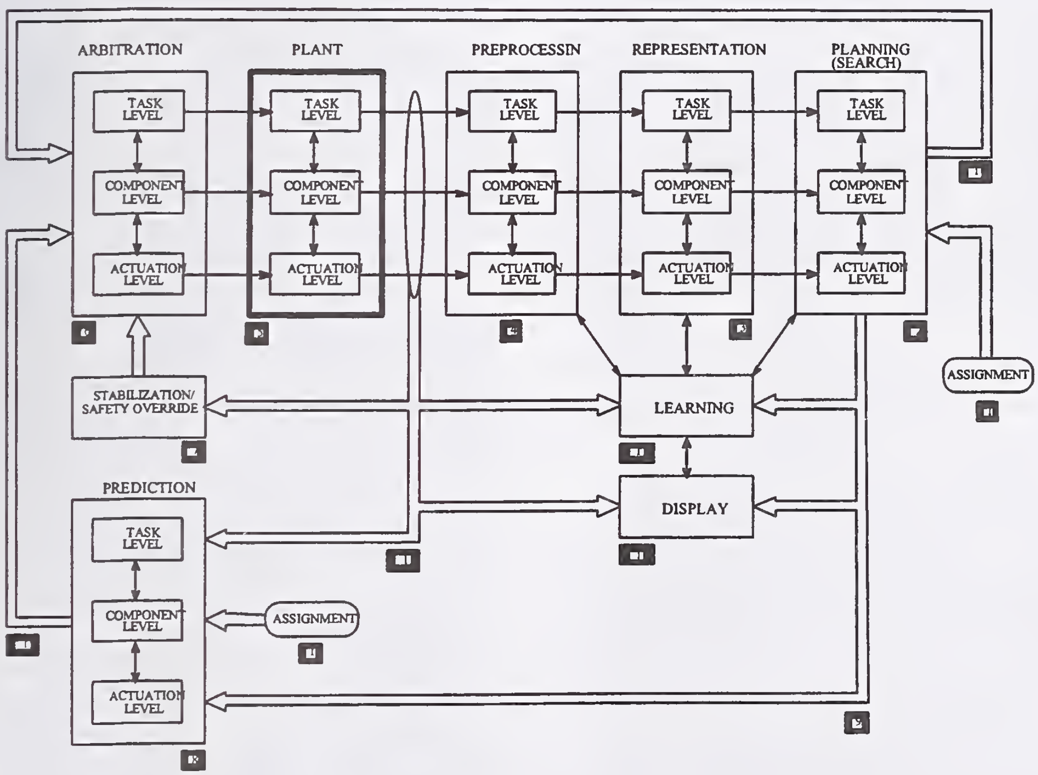


Figure 3. Architecture of the Power Station Multiresolutional Decision Support System for energy efficient Planning/Control

Figure 3 contains several subsystems that should be added to the standard ELF if there is a need to equip the system by Multiresolutional Decision Support: these are the systems of Learning and Prediction. Learning enhances Representation, while Prediction, together with Stabilization arrives at the subsystem of Arbitration (6). Standard ELF is the core structure demonstrating the important property of any functioning system, including Intelligent Systems: it shows the property of *closure*. The meaning of closure is in the fact that the proper functioning requires the loop of information flows to be closed.

What happens in the subsystem of Behavior Generation? The latter has two mechanisms: 1) maintenance of the ontology that organizes the inventory of the known symbols and their definitions and keeps relationships of nestedness with ontology subsystems of other resolution levels, 2) combinatorial search engine that performs *planning* i. e. creates alternatives of imaginary (possible and desirable) worlds, and 3) *simulator* engine that explores expected behaviors of the alternatives of the imaginary worlds and/or monitors the *execution* processes.

The goal of functioning to be achieved by the system arrives at the subsystem of Behavior Generation [9] that is equipped by mechanisms of planning and execution. At the present time, these mechanisms cannot be considered as thoroughly studied, and the general theory of planning can hardly be attempted. We will discuss a subset of problems in which the goal is defined as the attainment of a particular state or a particular string of states. Other types of problems can also be imagined: in chess the goal is clear -to win but this goal demand achieving a special configuration (mate-situation) but it cannot be achieved by arriving at a particular pre-determined position in a space (even in a descriptive space.) Most of the problems related to the theory of games and linked with pursuit and evasion are characterized by a similar predicament and are not discussed here.

Planning is understood as searching for appropriate future trajectories of motion leading to the goal. Searching is performed within the system of representation (simulation) that gives a tremendous advantage in comparison with searching via trying.

3. Planning in a Representation Space with a Given Goal

The world is assumed to be judged upon by using its Space of Representation, or its State Space which is interpreted as a time tagged vector space with a number of properties. Any activity in the World (State Space · or Representation Space) is called *motion*. It can be characterized by a trajectory of motion with the "working point" or "present state" (PS) that is traversing the space from one point (initial, or state, IS) to one or many other states (goal states, GS.) The goal states are given initially from the external source as a "goal region", or a "goal vicinity" in which the goal state may not be completely defined in a general case. This vision of the problem of Behavior Generation was dominating in the area of Control Systems. Planning was unified with Control only recently when it became clear that both Planning and Control are involved into anticipation of the preferable motion (off-line) with some appropriate correcting activities (on-line).

One of the stages of planning (often the initial one) serves for defining where exactly is the GS within the "goal region." In this paper, we will focus upon a subset of planning problems where one or many GS remain unchanged through all period of their achievement. Traversing from IS to GS is associated with consuming time, or another commodity (performance index, or cost.).

Planning Problems in Behavior Generation is frequently associated with the domain of robotics or automated control systems although it is absolutely equivalent to planning in all other domains. Robotics became the integrated research domain that provided for blending the goals and testing the means of achieving them, i.e. a domain with a direct need for planning. In 1983, T. Lozano-Perez has introduced the idea of search in "configurations space". From the experience of using this search, it became clear that the exhaustive search would be computationally prohibitive if the configuration space is tessellated with the final accuracy required for motion control. The theory of configuration space made one important thing obvious: planning is searching for admissible alternatives. This development helped to realize that planning should combine the exhaustive (often meaningfully complex) search off-line, and an efficient algorithm of an off-line control.

At this period of time the engineering community stopped talking about control of actions and introduced a more balanced term of Behavior Generation

The overview of the situation in the area of planning and control can be found in [9]. The recommended algorithm should be aligned with the following suggestions. Consider the Ω state space in which the start and final points SP and FP are given. The minimum cost path from SP to FP is to be found with the final accuracy ρ. Let us consider particular cases $\Omega = \Omega_1$ and $\rho = \rho_m$. To declare the final accuracy is equivalent to applying some mechanism od space tessellation. One of mechanisms of tessellation is distributing discrete points in the state space. We will distribute them in a random fashion and then, will determine the minimum cost path while considering these points as vertices of an imaginary graph. The condition of constructing random tessellation reflects uncertainties of the system that should be available for evaluation from the existing representation. In Figure 4,a the random points are distributed in the state space with obstacles. An example of the result of running a minimum-cost algorithm in the tessellated state space with obstacles is shown in Figure. 4,b.

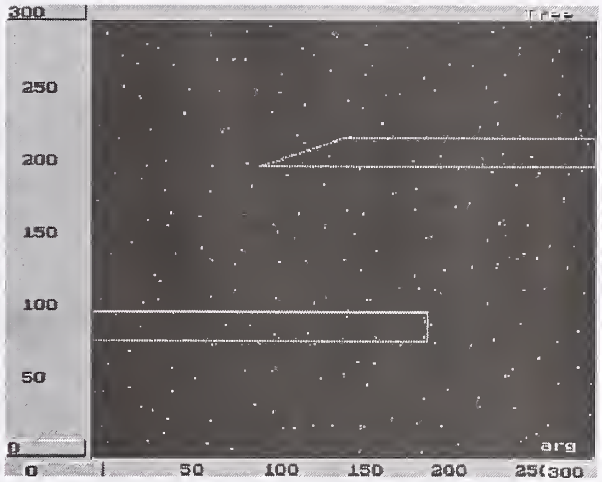
Since the graph is randomized, the trajectory is a random one, too. If one runs the search algorithm a number of times, we receive the results of searching as a "stripe of solution" as shown in Figure 5,a. Then, we get a privilege to continue with constructing tessellations of higher resolution only within this stripe as shown in Figure 5,b. Then, when we run the minimum cost search-algorithm only within this "finely" tessellated stripe with high resolution density of tessellata, we receive a high resolution plan. This process can be repeated recursively within as many levels as necessary.

We will introduce three operators that describe the above computations.

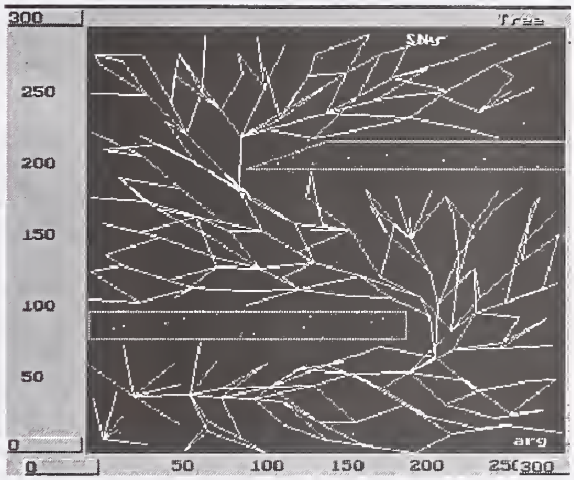
I. Operator of Representation ($\mathfrak{R}$)

$$\mathfrak{R}:(\Omega,\rho)\rightarrow M, \text{ or } M=\mathfrak{R}(\Omega,\rho) \quad , \qquad (1)$$

where M- is the map representing the state-space Ω, ρ is the level of resolution of this map determined by the density of the search-graph that we intend to run at this particular level of resolution (determined by the accuracy ρ). This is a non-trivial operator because it presumes discovery of entities, putting them into relationships with each other, generalization, instantiation and measuring relationships (including costs).



a



b

Figure 4 Randomized tessellation of the state space (a) and a single running of the minimum cost trajectory algorithm (a rectangle and a trapeze – obstacles)

II. Operator of state space search (S^3)

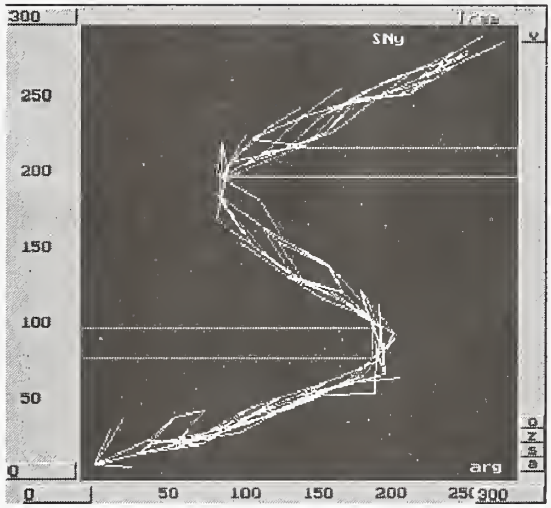
$$S^3: (M, SP, FP, J, \rho)\rightarrow P \quad , \qquad (2)$$

or $P=S^3(M)$,where P- is the optimum path connecting the start point SP and the finish point FP with tessellation constructed for the accuracy ρ. J- is the cost of operation which should be minimized as a result of search S^3. This operator should be based upon one of the minimum-cost algorithms (e.g. Dijkstra) and tailored for specifics of the problem.

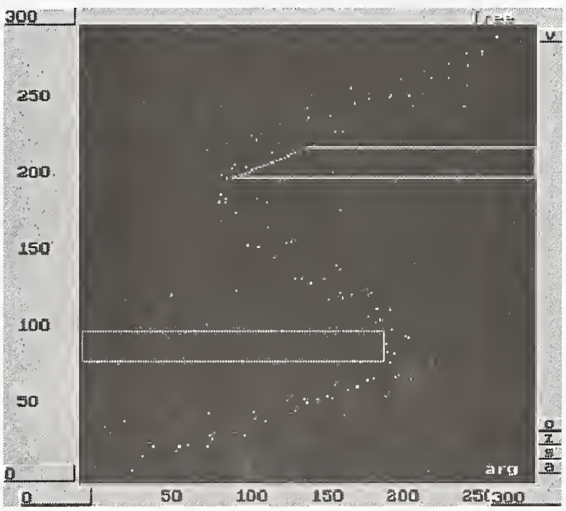
III. Operator of space contraction (C) which determines the width of stripe and the new final goal for the algorithm of search.

$$C:(P, w)\rightarrow\Omega, \text{ or } \Omega=C(P), \qquad (3)$$

where w- is the width of the "stripe" obtained after several runs of the search algorithm. Instead of "stripe" one can use the term "envelope", (e.g. the "width" is the parameter of the envelope).



a



b

Figure 5 Multiple running of the minimum cost trajectory algorithm (a) and the uncertainty stripe obtained as a result of the multiple running (b)

The hierarchical control algorithm can be described as follows:

for k=1, ..., m do the following string of procedures:

a) $\Omega_k = C(P_{k-1})$, or at k=1 assume $\Omega_k = \Omega$,

b) $M_k = \Re(\Omega_K, \rho_k)$,

c) $P_k = S^3(M_k)$.

The algorithm of control can be represented as a diagram

$$
\begin{array}{ccccccc}
 & & w & & \rho_k & & SP, FP, J \\
 & & \downarrow & & \downarrow & & \downarrow \\
P_{k-1} & \rightarrow & X & \longrightarrow & P & \longrightarrow & \Sigma^3 \longrightarrow P_k \quad (4) \\
 & & & & \uparrow \Omega_k & & \uparrow M_k
\end{array}
$$

or a recursive expression

$$P_k = S^3(R\ (C\ (P_{k-1}, w), \rho_k)\ SP, FP, J) \qquad (5)$$

The algorithm (5) has proven to be good for off-line search in the state space. In the class of on-line problems the process of control is to be described by the trajectory of "working point" moving in the state space.

4. Learnable Representations

All Representation Spaces are acquired from the external reality by the processes of Learning. Many types of learning are mentioned in the literature (supervised, unsupervised, reinforcement, dynamic, PAC, etc.). We will focus primarily on processes of unsupervised learning [12]. Before classifying the needs for a particular method of learning and deciding how to learn, we would like to figure out what should we learn. Now, it is not clear whether the process of learning can be separated algorithmically into two different learning processes: a) of *objects representation*, and b) of the *rules of action representation*, or are these two kinds of learning just two sides of the same core learning process. In both cases, learning is storing and generalizing information of experiences with their values associated with achieving particular goals.

The following knowledge should be contained in the Representation Space. If no GS is given, any pair of *state* representations should contain implicitly the *good rule* of moving from one state to another. In this case, we consider any second state as a provisional GS.

We will call "proper" representation a representation similar to the mathematical function and/or field description: at any point of the space, the derivative is available together with the value of the function. The derivative can be considered an action required to produce the change in the value of the function.

We will call "goal oriented" representation a representation in which at each point a value of the action is given required for describing not the best way of achieving an adjacent point but the best way of achieving the final goal. Both "proper" and "goal oriented" representations can be transformed in each other. Neither is mandatory for functioning: valued *memories of experiences* (ME) are sufficient.

5. The Artifacts of Representation Space: The Phenomenon of "Sea Weeds"

Representation as sets of valued ME is characterized by the following artifacts:

- existence of states with its boundaries determined by the resolution of the space each state is presented as a tessellatum [9], or an elementary unit of representation, the smallest discernible unit of attention)
- characteristics of the tessellatum which is defined as an indistinguishability zone; we consider that resolution of the space shows how far the "adjacent" tessellata (states) are located from the "present state" (PS)
- lists of coordinate values at a particular tessellatum in space and time
- lists of actions to be applied at a particular tessellatum in space and time order to achieve a selected adjacent tessellatum in space and time
- existence of strings of states intermingled with the strings of actions to receive next consecutive tessellata of these strings of states
- boundaries (the largest possible bounds of the space with similar properties, i. e. the obstacles
- costs of traversing from a state to a state and through strings of states.

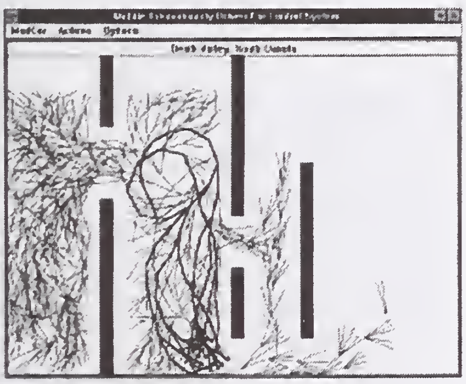


Figure 6. The sea weeds

When ME are clustered into classes of similarity (e.g. of adjacency) they remind visually masses of "sea weeds". In many cases, the states contain information pertaining to the part of the world beyond our ability to control it, and this part is called "environment." The part of the world to be controlled is the system for which we plan often referred to as "self." Thus, the representation is a part of "self" including knowledge about actions that "self" should undertake in order to traverse the environment. Plans are formed as strings of preferable "sea weeds" combined together.

6. Planning in Redundant Systems

Non-redundant systems have a unique trajectory of motion from one state to another. Redundant systems are defined as systems with more than one "the best" trajectory of motion from initial (IS) to final states (FS)

These systems contain a **multiplicity of alternatives of space traversal**. Redundancy grows when the system is considered to be a stochastic one. The number of available alternatives grows even higher when we consider also a multiplicity of goal tessellata at a particular level of resolution. This happens when the goal is being assigning at a lower resolution level which is the fact in multiresolutional systems (such as NIST-RCS [8, 9])

In non-redundant systems, there is no problem of planning. The problem is to find the unique trajectory and to provide tracking of it by an appropriate classical control system.

7. Learning as a Source of Representation: Storing and Clustering "Sea Weeds"

Learning is defined as knowledge acquisition via experience of functioning. Thus, learning is development and enhancement of the representation space. The latter can be characterized in the following ways:

- by a set of paths (to one or more goals) previously traversed
- by a set of paths (to one or more goals) previously found and traversed
- by a set of paths (to one or more goals) previously found and not traversed
- by a totality of (all possible) paths
- by a set of paths executed in the space in a random way.

One can see that this knowledge contains implicitly both the description of the environment and the description of the actions required to traverse a trajectory in this environment.

All information arrives in the form of experiences. The "learned" representation as a set of strings of valued "sea weeds" is equivalent to the multiplicity of explanations how to traverse, or how to move.

8. Types of Problems of Planning

Any problem of planning is associated with

- actual existence of the present state
- actual, or potential existence of the goal state
- knowledge of the values for all or part of the strings of executable states as far as some particular goal is concerned.

Any problem of planning contains two components: to refine the goal (bring it to the higher resolution) and to determine the path to this refined goal. They are performed together, or separately and can be formulated as follows:

a) given PS, GS and KS find the subset of KS with a minimum, or a prearranged cost, or with a cost in a particular interval.

b) given PS, Gs from the lower resolution level and KS (all paths) find the GS with a particular value

Finding solutions for these problems is done by a process of *planning*. In other words, planning is a construction of the goal states, and/or strings of preferable states connecting the present state with the goal states. There is a striking similarity and interrelatedness between *planning* and *learning*, actually their inseparability.

In order to do this, we must learn where the goal is located by consecutive refinement of the initial coarse information. In all cases it is associated with reduction of the indistinguishability zone and the size of the tessellatum associated with a particular variable, i.e. the accuracy of representation grows. We plan and learn by testing: in the representation, for planning, and in the reality, for learning. Learning via testing simulated systems is becoming more and more wide spread.

The second component is the simulation of all available alternatives of the motion from the initial state. Procedurally, this simulation is performed as a search, i.e. via combinatorial construction of all possible strings (groups). To make this combinatorial search for a desirable group more efficient we reduce the space of searching by focusing attention.

The need in planning is determined by the multialternative character of the reality The process of planning can be made more efficient by using appropriate heuristics which are available via processes of learning.

9. The Unified System of Planning and Learning: A Subsystem of MR DSS

The process of searching for plans is associated either with collection of additional information about experiences, or with extracting from KS the implicit information about the state and moving from state to state, for the purpose of learning. In other words, ***planning is inseparable from and complementary to learning***.

This unified planning/learning process is always oriented toward improvement of functioning in engineering systems (improvement of accuracy in an adaptive controller, improvement of efficiency in energy consuming devices) and/or toward increasing of probability of survival (emergence of the advanced viruses for the known diseases that can resist various medications, e.g. antibiotics.)

This joint process can be related to a system as well as to populations of systems and determines their evolution.

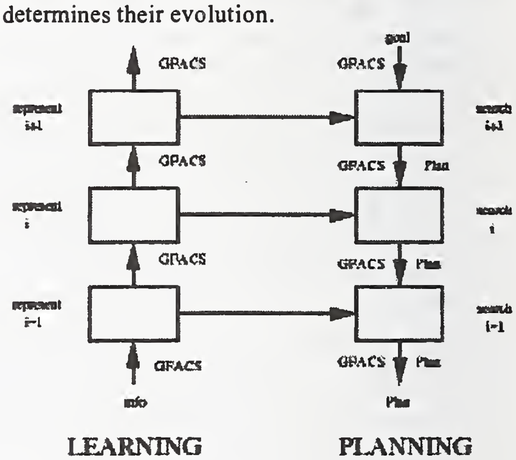


Figure 7. On the relations between planning and learning

10. Planning, Learning, and Control: A Unified Theory

Learning/Planning Automaton. The joint Planning/Learning process is studied by using a tool of Learning/planning automata (LPA) is a tool that allows for jointly exploring these two fundamental processes of intelligent systems. Naturally, it becomes a component of the Multiresolutional DSS.

Elementary Computations. *Search* (S) is always performed by constructing feasible combinations of the states within a subspace ("feasible" means: satisfying a particular set of conditions or constraints.) As many as possible alternatives of feasible motions should be explored and compared. If search is combined with formation of alternatives, we call this procedure *combinatorial search* (CS).

Usually, *grouping* (G) presumes exploratory construction of possible combinations of the elements of space and as one or many of these combinations satisfy conditions of "being an entity", this group generates a new symbol with subsequent treating it as an object.

The larger the space of search is, the higher is the computational complexity of search. This is why a special effort is allocated with reducing the space of search, i.e. focusing attention (FA) upon reduced sub-spaces. FA results in determining two conditions of searching, namely, its upper and lower boundaries:

a) the upper boundaries of the space where the search is to be performed, the *scope*

b) the resolution of representation (the lower boundaries, the tesselatum)

Via exploring these experiences in planning and learning we arrive at a conclusion that they are always employ these three procedures: grouping, focusing attention and combinatorial search (or subsets of them).

The property of Intelligence. Forming multiple combinations of entities (combinatorial search, CS) satisfying required conditions of transforming them into new entities (grouping, G) within a bounded subspace (focusing attention, FA) is frequently performed as a fundamental set of procedures. Since these three procedures work together we will talk about them as about a triplet of computational procedures (the abbreviations GFACS or CFS are used.) Notice, that in learning it creates lower resolution levels out of higher resolution levels (bottom-up) while in planning it progresses from lower resolution levels to higher resolution levels (top-down). This algorithmic triplet emerges as a tool of multiresolutional representation and/or for the purposes of generating goal-oriented behaviors.

This triplet of computational procedures is characteristic for *intelligence* of living creatures and constructed systems, and probably is the elementary computational unit for

The need in GFACS is stimulated by the property of knowledge representations to intelligence. Its purpose is transformation of large volumes of information into a manageable form that ensures the success of functioning. This explains the pervasive character of hierarchical architectures in all domains of activities including Decision Support Systems.

contain a multiplicity of alternatives of space traversal (i. e. a property of any representations to be **redundant**.) Representations reduce the redundancy of reality. This allows for having problems that can be solved in a closed form (it is a form when no combinatorics is possible and/or necessary).

At each level of resolution, planning is done as a reaction for the slow changes in situation which invokes the need in anticipation and active interference

a) to take advantage of the growing opportunities, or

b) to take necessary measures before the negative consequences occur.

The deviations from a plan are compensated for by the compensatory mechanism also in a reactive manner. Thus, both *feedforward* control (interpreted as planning at all levels of resolution but the highest one) and *feedback* compensation of deviations are reactive activities. Both can be made active in different implementation approaches in control theory.

Examples: a) Classical control systems are systems with no redundancy, they can be solved in a closed form without searching.

b) Any stochastic condition, any type of uncertainty introduced to a control system creates redundancy and requires either for elimination of redundancy or performing search.

c) Optimum control allows for the degree of redundancy that makes searching feasible.

In Figure 8, the process of multiresolutional planning via consecutive search with focusing attention and grouping is demonstrated for the control problem of finding a minimum-time motion trajectory. From Figure 8, one can judge the processes that are performed during the single level S^3-search in the randomized tessellated state space. The reader can identify the processes because a trivial example is considered: minimum time functioning of the dynamic system. The operation won't change if one is dealing with higher order and/or non-linear system.

The space is learned and encoded in advance by multiple testing, and its representation is based upon knowing that the distance, velocity and time are linked by nonredundant expression. Several methods of constructing attention envelopes are applied.

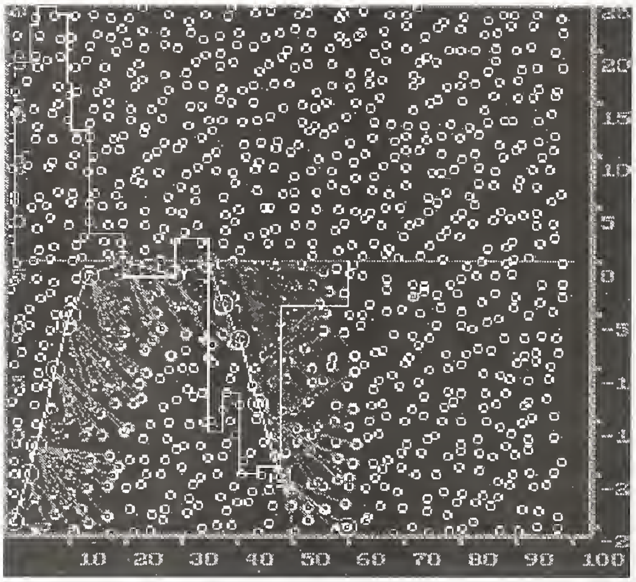


Figure 8. S^3-Search in the state space for the minimum tyme dynamic trajectory

Conclusions

We have demonstrated the advantages of Multiresolutional Decision Support Systems that can be listed as follows:

1. Multiresolutional System of Knowledge Organization allows to reduce complexity and increase efficiency of representation.

2. Most of the Planning/Control problems are being solved via S^3-search (Search in the State Space). The latter requires performing randomized state space tessellation with density of points that reflects the uncertainty of information. Multiresolutional S^3-search allows for stochastic optimization of systems.

3. Representations at each level of resolution are organized as memories of experiences and do not require constructing any analytical model: this system plans and controls with no model required.

4. This representation supports processes of unsupervised learning that contains self-oriented information; no special self-identification is required.

5. The MRDS system was tested in applications to power station energy efficient planning/control system, for planning/control of an unmanned autonomous mobile or spray-casting robots.

References

1. A. Satyadas, U. Harigopal, N. P. Cassaigne, "Knowledge Management Tutorial: An Editorial Overview," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, Vol. 31, No. 4, November 2001, pp. 429-437
2. P. Morrison and P. Morrison, and the Office of C. and R. Eames, *Powers of Ten*, Scientific American Library, A division of HPHLP, NY, 1982
3. A. Meystel, "Planning in a hierarchical nested controller for autonomous robots," Proc. IEEE 25th Conf. on Decision and Control, Athens, Greece, 1986, pp. 1237-1249
4. A. Meystel, "Theoretical Foundations of Planning and Navigation for Autonomous Mobile Systems", *International Journal of Intelligent Systems,* Vol.2, No.2, 1987, pp. 73-128
5. C. Isik, A. Meystel, "Pilot level of a hierarchical controller for an unmanned mobile robot," *IEEE J. of Robotics & Automation*, Vol. 4, No. 3, 1988, pp. 244-255
6. A. Meystel, Autonomous Mobile Robots, World Scientific, World Scientific, Singapore, 1991
7. A. Meystel, "Nested Hierarchical Control," in Eds. P. Antsaklis, K. Passino, *An Introduction to Intelligent and Autonomous Control*, Kluwer Academic, Boston, 1993, pp. 129-161
8. A. Meystel, "Architectures, Representations, and Algorithms for Intelligent Control of Robots", in *Intelligent Control Systems: Theory and Applications*, eds. M. M. Gupta, N. K. Sinha, IEEE Press, New York, 1996, pp. 732-788
9. A. Meystel, J. Albus, *Intelligent Systems: Architecture, Design, Control*, Wiley, NY., 2001
10. J. Albus, A. Meystel, *Engineering of Mind: An Introduction to the Science of Intelligent Systems,* Wiley, NY, 2001
11. C. Corson, A. Meystel, F. Otsu, S. Uzzaman "Semiotic Multiresolutional Analysis of a Power Plant", ", in Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems, Proc. of the 1995 ISIC Workshop, Monterey, CA, pp. 401-405
12. A. Meystel, "Learning Algorithms Generating Multigranular Hierarchies", in *Mathematical Hierarchies and Biology*, Eds.: B. Mirkin, F. R. McMorris, F. S. Roberts, A. Rzhetsky, DIMACS Series in Discrete Mathematics, Vol. 37, American Mathematical Society, 1997

Motivated Metamodels

Paul K. Davis
Senior Scientist, RAND and Professor,
RAND Graduate School
1700 Main St. Santa Monica, CA 90407
(pdavis@rand.org)
Santa Monica, CA 90407
(pdavis@rand.org)

James H. Bigelow
Senior mathematician, RAND
1700 Main St., Santa Monica, CA
90407-2138
(James_Bigelow@rand.org)

ABSTRACT

A metamodel is a relatively small, simple model that approximate the "behavior" of a large, complex model. A common way to develop a metamodel is to generate "data" from a number of large-model runs and to then use off-the-shelf statistical methods without attempting to understand the model's internal workings. It is much preferable, in some problems, to improve the quality of such metamodels by using various types of phenomenological knowledge. The benefits are sometimes mathematically subtle, but strategically important, as when one is dealing with a system that could fail if any of several critical components fail. Naïve metamodels may fail to reflect the individual criticality of such components and may therefore be misleading if used for policy analysis. By inserting an appropriate dose of theory, however, such problems can be greatly mitigated. Our work is intended to be a contribution to the emerging understanding of multiresolution, multiperspective modeling.

Keywords: *metamodel, multiresolution modeling, model abstraction, response surfaces, repro models, statistics, regression, intelligent machines, robotics, machine planning*

INTRODUCTION

A metamodel is a relatively small, simple model intended to mimic the "behavior" of a large, complex model. Two reasons for wanting to build metamodels are.[1]:

- Cognitive. We want to "understand" why the large model behaves as it does. This will enhance the model's meaningfulness and credibility with ourselves, other analysts, and with whomever we seek to influence.
- Exploratory analysis. We often want to explore the behavior of a model over a large part of its domain. A metamodel with only, e.g., 5-10 (rather than hundreds or thousands) of variables makes comprehensive exploratory analysis feasible and comprehensible[2], [3].

Sometimes, it is possible to build models using multiresolution modeling design principles [1], in which case the low-resolution versions (the more abstracted or aggregate versions) already have these virtues. Often, however, the baseline situation is that as ubject area has been modeled only in detail. Often, the detailed model is old, opaque, and difficult to work with. A metamodel, then, is an attempt to generate a low-resolution version after the fact.

Consider first two extreme approaches to building a metamodel. For the *statistical* approach, one runs the large model many times for a variety of inputs; one then collects the inputs and outputs in a big dataset. A statistician analyzes these data as he would data collected in any series of experiments. He seeks a statistical model that does a good job of estimating outputs, while using as few input variables as possible so as to keep the model simple and so as to avoid "overfitting" his initial data.

An idealized *phenomenological* (or theory-driven) approach starts with the most exact theory available and derives a simplified model by rigorously aggregating (as in replacing integrals over volume with a representative value times the volume), rearranging and combining terms and factors, and so on—using physical insights wherever possible (e.g., recognition of some conservation principles or of being able to view a factor as an idealization times an efficiency factor).

Some statisticians, operations researchers, and computer scientists prefer the first approach and want to know nothing about the "innards" of the model whose behavior they are attempting to replicate. They may have a purist philosophy of "allowing the data to speak," without "contaminating it" with theoretical assumptions. Or they may simply prefer not having to deal with the complexities of the model's innards: they may wish to turn the problem over to automated software. At the other extreme, some theoretically inclined academicians clearly prefer the second approach because it allows rigorous tying together of phenomena at different levels of detail (as when classical thermodynamics is understood from quantum mechanics). These, then, are the extremes. Most scientists, engineers, and analysts, however, should prefer something in between. Often, the "in-between" amounts to an analyst postulating some simple scaling relation and using data to calibrate that scaling relation. The scaling relation may be naïve and the calibration may use only the crudest of statistics.

Our interest in this research was to clarify principles for doing better. We had some ideas for how to do so, based on theoretical reasoning, but we preferred where possible to test and iterate ideas by experimenting with specific well-posed problems. With that in mind, we used a well-documented model that we could consider "large and complex model" and applied our ideas in stages, starting with pure statistical metamodeling and moving toward more theory-informed work. In what follows, we first discuss how one should judge the quality of a metamodel. We then describe our experiments and conclusions. The discussion adds to an earlier preliminary discussion [4] and is drawn from a longer and more technical piece [5].

WHAT MAKES A METAMODEL GOOD

We suggest five criteria for assessing the goodness of a metamodel, heroically assuming for the sake of this discussion only that the base model is in fact accurate:

Goodness of fit. Obviously, we want the metamodel's predictions to be reasonably consistent with those of the baseline model. A measure of this is the root mean square error (or fractional error) of predictions across the domain of input values. This is superior to the commonly used R^2.

Parsimony. For purposes of both cognition and exploratory analysis, a good metamodel will have relatively few independent variables, which ideally, would also be meaningful. Achieving parsimony may be accomplished by omitting some of the baseline model's inputs (i.e., treating them as constant) or by combining several base model inputs into a smaller number of intermedate variables. The set of independent variables should be rich enough to represent the issues being addressed with the model. Beyond that, the fewer extra variables, the better.

Identification of "critical components." Our third criterion seems new and we believe it to be crucial. Many uses of models in analysis involve systems or strategies, the failure of which is to be very much avoided. We suggest that a metamodel should highlight all of the input variables that are essential to success—especially when troublesome values of those variables are plausible. The model should not give the impression that one can compensate for a weak component of the system by improving some other component (if such substitution is in fact inadequate). This is a significant consideration in metamodeling, because standard statistical methods lead to linear sums that imply substitutability. We refer to components that must individually succeed (have values above or below an appropriate threshold) as *critical* components. If critical components in this sense exist, the metamodel should be appropriately nonlinear.

Reasonable depiction of relative "importances." Metamodeling can generate statistical measures of the significance or importance of candidate variables. In stepwise regression, the less significant variables are dropped. That, in turn, could mean that someone using the metamodel for resource allocation would consider the dropped variables as unimportant. A good metamodel would give no misimpressions on this score.

A good storyline. Without a story, a model is just a "black box." A story explains *why* the model behaves as it does. More, it relates the model to the real world, telling us why the model *should* behave as it does. We use the term "storyline" because all models are a simplification of reality, but we intend no cynicism. Said differently, the model should be "physically (or otherwise phenomenologically) meaningful and interpretable," not just a math formula.

With this background defining what we mean by a good metamodel, let us now describe the analytical experiments we conducted to illustrate and sharpen our understanding of ways to improve metamodeling.

THE EXPERIMENT

Our experiment was to begin was a relatively large and complex model and to develop a series of metamodels to represent it. For the first metamodel we relied almost entirely on statistical methods, uninformed by phenomenology (i.e., our knowledge of the workings of the base model). With each successive metamodel, we took advantage of progressively more phenomenology.

The Large Model

Our "large, complex model" (i.e., our baseline model) was EXHALT-CF [5],[6], which treats the so-called halt phase of a military operation. Although much simpler than real base models of interest, it has scores of variables and a great many nonlinearities. It seemed complex from its documentation and program.

In its simplest version, the halt phase is a mere race. An attacking force (Red) is advancing on an objective while the defenders (Blue) interdict its armored vehicles with long-range fires. Red will halt when he reaches his objective (a Red win) or when Blue has killed a specified number of vehicles (a Blue win), whichever comes first. EXHALT-CF, however, adds many embellishments relevant to current strategic concerns about real-world military operations, especially in the Persian Gulf.

First, the model must represent Blue deployments. Some number of shooters may be stationed in theater in peacetime. Depending on strategic warning, diplomatic relations, Red's deceptiveness, and Red's ability to threaten bases in theater (e.g., with weapons of mass destruction), Blue may or may not be able to augment this number before Red begins his advance. Once Red's advance begins, Blue will deploy more shooters into the theater, up to a theater capacity, which reflects logistical shortcomings.

The effectiveness of Blue shooters is measured by kills per shooter-day. Early in the campaign, Blue may be unable or unwilling to attack the Red column because of Red air defenses. After a period of air-defense suppression, Blue's attacks will start. Even then, however, sortie rates may be reduced because of a continued threat of attack with mass-destruction weapons, which would force Blue personnel to work in protective gear or would force Blue to operate from more distant, and more poorly prepared bases.

The weapons and strategy Blue selects will also influence Blue shooter effectiveness. Blue may select an area weapon, capable of killing several Red armored vehicles per shot. To counter this, Red may space his vehicles more widely. Or Blue may select a point weapon, which kills no more than one vehicle per shot, and is unaffected by Red's vehicle spacing. Also, Blue will likely have limited supplies of his best weapons, and revert to lesser weapons when his best are exhausted. Blue may attack the entire Red column in depth (the "In Depth" strategy) or focus his attack on the leading edge (the "Leading Edge" strategy). If Blue does the latter, his attack may slow Red, but each sortie may be less effective due to deconfliction problems.

These and other complications of the halt problem are represented in EXHALT-CF and the simulation version, EXHALT. They are implemented in Analytica™, a graphical modeling environment for the personal computer. EXHALT-CF has 63 inputs, 8 switches to turn features on or off (the model has a multiresolution, multiperspective design) , three indexes, and 451 variables that are calculated directly or indirectly from inputs. For our purposes, we focused on a subset of cases, which reduced to 25 the number of input variables affecting the problem. This seemed adequately complex to illustrate our points—or, more accurately, to allow us to experiment. The experiments in question were experiments of discovery and learning, not rigorous hypothesis testing.

The Experimental Data

We selected statistical distributions, mostly uniform distributions, from which to generate the 25 variable-value inputs. We then ran EXHALT-CF to generate a Monte Carlo sample of 1000 cases from the overall input space. We did not weight one or another region of the input space because we were seeking a broadly good fit of behavior over the entire domain of interest.

METAMODELING

Metamodel 1

In our first experiment, we acted as though we had handed the dataset to a statistician (or statistically oriented operations researcher or computer scientist), and commissioned him to develop an estimator for the halt distance that Blue could achieve using his best strategy and weapon type for the circumstances of the case. A good statistician would insist on discussing the problem. He would want to know which data elements to use as independent variables, which are outcome variables, and so on. Even if he preferred to operate as thought the original model is a "black box," he would probably want at least some interpretation of the variables' meanings. This would permit him to do some data manipulation that would simplify his analysis.

For his initial analysis, our simulated statistician specified a linear model with 25 independent variables. Because he knew that we wanted a parsimonious metamodel, he ran a *stepwise linear regression* procedure in which the independent variables were added to the model one by one in the order of decreasing explanatory power. That is, the first variable considered yielded the largest reduction of the root mean square of the residual error (RMS Error. After the first six or seven variables, further additions didn't improve the fit very much. Actually, the fit wasn't very good no matter how many variables were included (the standard error was on the order of 25%, which in this problem is large).

In any case, our simulated statistician stopped with 14 variables, all coefficients of which were significant at the 0.05 level.

How good was metamodel 1? Earlier we identified five features that make a metamodel good. The performance was not impressive, although, in our experience with metamodeling pure statistical approaches such as this not uncommonly do quite well by the average goodness-of-fit criterion. Still, in this case, there were 421 cases in which Red actually reached his objective and the model estimated that Red was halted short of his objective in 92 percent of them.

Parsimony was our second criterion for a good metamodel. This model had 14 variables. We would like fewer, but this was perhaps a marginally acceptable number.

Identification of critical components was our third criterion. Here Metamodel 1 performed very poorly. Such metamodels are linear in the variables identified as significant by the statistical analysis. Thus, when used, metamodel 1 failed to identify and highlight critical components. For example, the model would predict that by merely improving munitions sufficiently, Blue could guarantee a small halt distance—independent of the other variables. That is flatly wrong. The model also gave a very misleading sense of the relative importance of variables. After stepwise regression, one variable had dropped out, while another—which entered the problem in precisely the same way (if the variables were X and Y, they entered the problem only through the product XY)—was retained as significant. This could be a very serious shortcoming if the model were used to inform resource-allocation decisions. Also, as discussed earlier, ur final criterion for a good metamodel was that it have a good storyline. This metamodel had no storyline at all.

Metamodel 2

A statistician will often try to improve his model by introducing transformations of the independent variables, such as exponentials, powers, and products of variables. That is, he will still use linear regression, but with some of the variables of that regression being nonlinear composites of others. So many possible transformations of variables are possible that the statistician may need some guidance selecting which ones to try. Brute force (e.g., considering all of the quadratic combinations of elementary variables as new, composite variables) can result in a good fit, but usually with even more statistically significant variables and no "story."

Phenomenology (i.e., the "innards" of the baseline model) can suggest what transforms to try, including transforms that statisticians do not generally consider. These include transforms that use the MAX and MIN operators. Indeed, a number of transformations are built into EXHALT-CF. We designed it as a *multi-resolution* model, to permit the user to specify inputs at different levels of detail. Even if the metamodeler finds EXHALT-CF as a whole to be big and complex, even early chapters of documentation, which deal with various idealizations, are sufficient to highlight natural composite variables. They may not *fully* substitute for the more elementary variables, because the "real" EXHALT-CF (as distinct from the simplified versions discussed in early documentation chapters) includes more complex interactions. Nonetheless, we thought that the suggested composite variables ("aggregation fragments") might go a long way.

For Metamodel 2, then, we looked at a number of such composites."

The simulated statistican then defined a linear model with far fewer variables, many of them the composites.

How good was metamodel 2? The performance, while considerably better than Model 1, was still not impressive. The standard error was perhaps 80 km, rather than 140 km (with interesting values being in the range 0-600).

On grounds of parsimony the model was better, since only ten independent variables proved to be statistically significant.

The good news was that the model did predict the critical-component phenomenon: we had identified enough of the key composite variables so that we could see importaant nonliearities. In particular, to obtain a good halt distance, Blue had to address three issues simultaneously (with no substitution). These involved the number of initial shooters, the earliest time at which Blue could begin attacking Red effectively, and the number of "shooter days" required for success (a functon of Red's size and Blue's effectiveness per attack mission). This important "system feature" stood out.

However, this metamodel still did not have a storyline, although the variables at least had more physical significance.

Metamodel 3

So far the simulated statistician had been combining the original, low-level variables into intermediate variables that we thought were reasonable on phenomenological grounds. We might characterize this as a "bottom-up" strategy. Now we turned to a "top-down" strategy. We viewed this as explicitly building in a storyline. It depended on an understanding of phenomenology, but it not require that the theory for describing that phenomenology be analytically tractable (e.g., solvable in closed form).

One piece of knowledge used was inferrable from even minimal documentation of EXHALT-CF, to wit that the model considered two different Blue strategies and took the better of the two results as the answer. The model did similarly in comparing two different classes of weapons. In the earlier metamodels, these branches and use of MIN/MAX operators was all buried, but in Metamodel 3 we built that same macro logic in. This meant that Metamodel 3 actually involved four metamodels, plus logic to compare results from each.

More important, we inserted physical reasoning in simplified terms. Upon thinking about the problem physically, we could reason that the halt time was just the time required to kill the requisite number of Red targets. However, that depended on the number of Blue shooters, which increased linearly with time (subject to some further constraints in the full model), the size of the Red force, and the per-shooter-day effectiveness of Blue. We could write a simple analytical equation for this—so long as we glossed over details and inserted averages. We estimated an integral by the product of a time duration and the average number of shooters within the interval (without knowing precisely how to calculate that average).

We then made a very crude estimate of this and other averages. The result was dimensionally correct and not absurd, but it was not intended to be accurate.

What we were looking for was *structural* form. This we postulated as the basis for Metamodel 3, although building in fudge factors to measure error in the form assumed.

How good was metamodel 3? Metamodel 3 fitted the data much better than either of the two previous metamodels. It was also much more parsimonious than the previous metamodels. It had only five significant variables and the fudge factors proved to be not very large. Thus, the "story" understandable from the highly simplified model that did violence to mathematics by, e.g., treating an integral as a product of a duration and an average value during the period, was largely correct.

The model also did well in predicting the critical-component phenomenon. The composite variables that made ths possible in Metamodel 2 were also present in Metamodel 3, but now with a better story. Nor were there any serious errors with respect to the relative importance of variables. All in all, results were rather good: building in the approximate structure had paid off handsomely.

Metamodel 4

The last metamodel that we considered pushed the analytic work further. Upon reflection, it was possible—in this particular problem—to do a much better job of estimating the average number of shooters present during the halt phase. This required nothing more profound than simple integration and relatively simple algebra. A good student of first year calculus would be able to do the problem without difficulty. If we inserted this knowledge, the resulting metamodel was exceedingly accurate—so much so that it was embarrassing. The results demonstrated that the complexity of EXHALT-CF was the result of essentials having been obfuscated by the programming. The subtleties and complications were simply not necessary when the numbers were crunched.

Interestingly, however, Metamodel 4 was *not* better than its predecessor by our criteria. Why? Because, in the process of inserting the improved solution, the structural form became more complicated and non-transparent. This obscured the story, making it impossible to actually put the model up on viewgraph and explain what was going on as, e.g., a product of three meaningful variables divided by a fourth, with a small error term reflecting the many simplifications involved. If we wanted clarity and insight, then Metamodel 3 was arguably better. We say "arguable," because with clever presentation one could hide some of Metamodel 4's complications.

SUMMARY AND LESSONS LEARNED

Our experiments confirmed our belief that much could be gained by combining virtues of statistical and phenomenological (theory-informed) approaches. They confirmed and give more precise arguments to our skepticism about approaching metamodeling as an exercise in pure data analysis, with the baseline model merely being a black-box generator of data. Although the experiment dealt with only a single baseline model, the insights appeared to us to be relatively general—at least for purposes of suggesting general cautions and approaches to consider. We intend no grand claims here, but those were (and are) our impressions.

In our experiments, the application of statistical methods uninformed by phenomenology did not produce a good metamodel. In part it failed because the data we were fitting describe a highly nonlinear surface. A linear model might fit locally, but never globally. Moreover, there was no guarantee that the regression coefficients would be a good guide to the relative importance of the independent variables. It was necessary to introduce nonlinear combinations of the low-level inputs in order to obtain a good fitting metamodel. Phenomenology could motivate the construction and selection of the appropriate nonlinear combinations. It would be very difficult to discover them from the data alone.

A linear model also failed because the data did not describe a smooth surface. Like many models, EXHALT-CF makes liberal use of MAX and MIN operators to make either-or choices. So the data described a surface with "kinks" in it. When one fits a kinky surface with regression, the coefficients obtained from regression don't need to make sense. The regression results tell us the average importance of inputs, but not *when* each one is important. We found it necessary to use phenomenology to separate the smooth segments of the surface, after which we could do a good job of fitting each smooth segment with just plain statistics.

As we introduced more and more phenomenology, we obtained successively better fitting models. We would argue that in addition, Model 1 had very little cognitive value (i.e., it didn't tell a story) and Model 2 had only a little more. Model 3, however, did tell a coherent story, one that could be related persuasively to the client. With Model 4 it could be argued that we began to lose cognitive value. The phenomenology became more complex and less transparent. All the equations began to obscure the story.

Particularly important also is that by inserting phenomenologically motivated structure one can avoid certain important blunders of system depiction. In particular, one can preserve and even highlight the role of critical components—components that enter the problem more nearly as products than as sums, or components that must individually have threshold values to avoid system failure. This is particularly important if metamodels are to be used in policy analysis or design. We would expect it to be quite important in design of intelligent systems, because we would expect it to be normal, not unusual, for designers to be worried about numerous independently critical factors.

ACKNOWLEDGMENTS

This research was sponsored by the United State Air Force Research Laboratory.

REFERENCES

[1] Davis, P. K. and J. H. Bigelow, *Experiments in Multiresolution Modeling*, RAND, Santa Monica, CA, 1998.
[2] Davis, P.K., J. H. Bigelow, and J. McEver, Exploratory Analysis and a Case History of Multiresolution, Multiperspective Modeling, RAND RP-925, Santa Monica, CA. Reprinted from *Proceedings of the 2000 Winter Simulation Conference*, J. A. Joines, R. R. Barton, K. Kang, and P.A.. Fishwick (editors), December, 2000 *and Proceedings of the SPIE*, Vol. 4026, 2000.
[3] Davis, P. K., J. McEver, and B Wilson, *Measuring Interdiction Capabilities in the Presence of Anti-Access*

Strategies: Exploratory Analysis to Inform Adaptive Strategy for the Persian Gulf, RAND, Santa Monica, CA MR-1471-AF, 2002.
[4] Davis, P.K. and J. H. Bigelow, "Meta Models to Aid Planning of Intelligent Machines," Proceedings of the 2001 PerMIS Workshop, September 4, 2001 (NIST Special Publication 982), Gaithersberg, MD.
[5] Bigelow, J.H. and P.K. Davis, "Developing improved metamodels by combining phenomenological reasoning with statistical methods," *Proceedings of the SPIE*, Vol. 4716, 2002 (A. Sisti and D A. Trevasani, editors).
[6] Davis P.K, J.H. Bigelow, and J McEver, *EXHALT: An Interdiction Model for Exploring Halt Capabilities in a Large Scenario Space*," The RAND Corporation, MR-1137-OSD, 2000

On the Role of Quality Attributes in Specifying Software/System Architecture for Intelligent Systems

Hessam S. Sarjoughian
Computer Science & Engineering
Arizona State University
Tempe, AZ 85287-5406
sarjoughian@asu.edu
www.acims.arizona.edu

Abstract

In recent years, researchers in software engineering have been exploring a range of issues on software and system architectures. One area that directly affects intelligent system is software and system architecture. Recent findings offer evidence that software architecture can be developed based on a set of well-defined concepts and methods. In this paper, we examine this line of research and its role in the field of intelligent systems. We suggest using *architecture lifecycle* as an approach to deriving architectures for intelligent systems. We introduce *intelligence* quality attribute and propose using it with a set of existing quality attributes to support incremental, evolutionary specification and evaluation of *intelligent software/system architectures.*

1 Introduction

The architecture of a computing system can be defined to be one or more structures composed of software/hardware components and their connections. A variety of architectural structures – conceptual, physical, control flow, data flow, module, call, process or coordination, and class – serve a collection of complementary, and sometime necessary, blueprints of complex systems. For example, a data flow structure allows one to trace satisfiability of functional requirement based on its constituting components and connections which represent programs and send data relationships respectively. These structures are of particular importance since they can serve as the first set of formal specifications suitable for "predicting" the desired behavior of intelligent systems. That is, a subset of such structures capture a system's behavior in terms of primitive artifacts that can be specified and qualified/quantified over a system's lifetime.

2 Software/System Architectures

The role of an architecture is to put in place a set of blueprints guiding the development of a system that can satisfy some desired behavior. The system's desired behavior can be formally specified in terms of a set of *quality attributes* such as performance and reusability. Many system architecture definitions, descriptions, and specifications have been proposed and employed from fields such as control theory, artificial intelligence, and software engineering (e.g., [1-9]). Bass, Clemens, and Kazman [4] offer a comprehensive account for the software/system architecture. Their treatment is founded on fundamental concepts (e.g., hierarchical structure) and techniques (e.g., modularization) proposed by Fred Brooks, Edsger Dijkstra, Kevin Iverson, and David Parnas among others. They present the architecture of a computing system as a set of specifications capturing its key elements in a systematic fashion. The approach allows one to specify and assess an architecture in a step-wise fashion – i.e., moving from domain independent to application-specific. The architecture, therefore, can be said to provide a set of artifacts that can be evaluated in terms of how well it supports or enables a system's desired external (e.g., availability and performance) and internal behavior (e.g., testability and modifiability).

More formally, an architectural specification of a computing system is defined as a structure or structures composed of software components, the externally visible properties of the components, and the relationships among them [4]. Each structure can be viewed as an skeleton of a system in terms of its components and connections. Each component has a type (software, hardware, or mixed), behavior (static, dynamic, or mixed), and roles (processing, control, mediating, etc.). The connections specify the type (communication, control, send/receive) and nature (e.g., asynchronous) of connections among the components. The structure itself specifies the topology of components and the kinds of interactions enabled – e.g., hierarchical, flat, layered, and mixed such as layered hierarchical.

The software/system architecture lifecycle[1] is shown in Figure 1a. These architectures can be derived in terms of three more primitive artifacts: architectural style, reference model, and reference architecture (see Figure 1). These artifacts provide "early design decisions". An architectural style specifies which quality attributes it supports based on generic description of component types and their patterns of runtime control and/or data transfer. The architectural style also specifies general constraints placed among components and types of interactions among them. For example, layered architecture defines how a layer may interact with layers below or above it and supports reusability quality attribute.

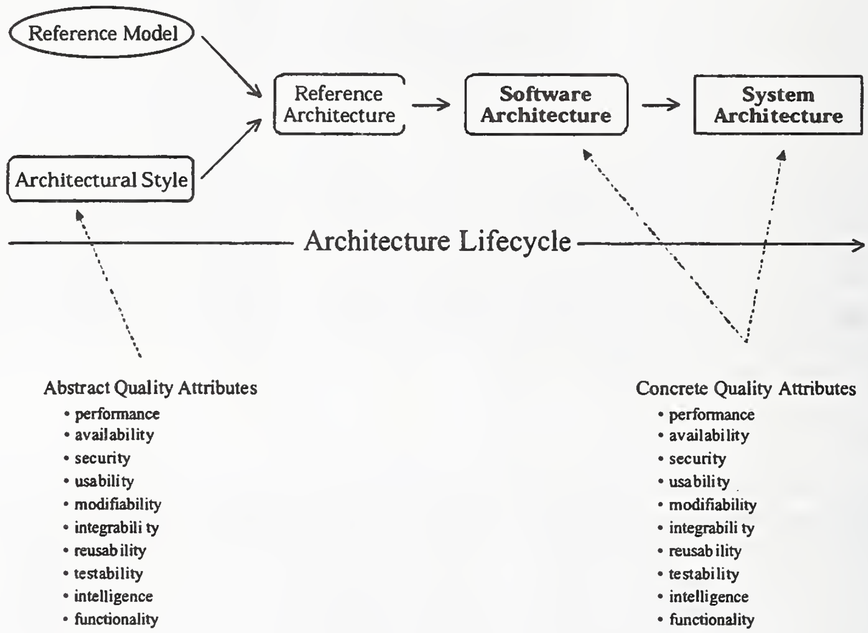


Figure 1: Architecture Lifecycle and Quality Attributes

The reference model, unlike architectural style, specifies a standard decomposition of a known problem (domain) into parts that cooperatively provide a suitable solution (e.g., Model-Viewer-Controller). A reference model represents a division of functionality together with data and control flow between them. The reference architecture is a reference model mapped onto components[2] along with their data and control flow where components cooperatively implement the functionality defined in the reference model. Next, we describe in some detail architectural style and quality attributes.

2.1 Architectural Styles

Architectural styles play a major role in the development lifecycle of intelligent systems since they can serve as the first step toward specifying and examining quality attributes (see Section 2.2). Unlike a software or system architecture, an architectural style specifies a family of architectures[3] – i.e., a set of components, and connectors along with a topology and its semantic constraints. In an architectural style, the topology specifies the layout of components and connectors as well as their collective interactions. Each component is designated to carry out some functions (e.g., a semi-autonomous vehicle processing sensory inputs). The connectors (e.g., procedure call, multicast, etc) are responsible for mediating communication, coordination, or cooperation among components. The architectural style,

---

[1] Lifecycle is iterative and evolutionary.

[2] Mapping of reference model to reference architecture is not necessarily one-to-one.

[3] It is important to note that architectural style not a kind of (software or system) architecture. Architectural style (a) does not specify the number of components, (b) dos not provide domain-specific, detailed functionality of the components and (c) does not give precise details for the interactions that may take place among components.

therefore, is an abstraction of a set of software/system architectures capturing their abilities and limitations from a domain-independent viewpoint. The importance of an architectural style is that it can be examined in terms of how well it enables some quality attributes (e.g., performance and integrability) and possibly impeding others (e.g., modifiability and security) in early stages of design.

2.1.1 Quality Attributes

Establishing quality attributes is an essential step toward specifying a system architecture which can facilitate or hinder some desired/necessary behavior. Quality attributes can serve as elementary artifacts for specifying and measuring a software/system architecture from distinct and complementary viewpoints. The quality attributes are categorized as runtime observable and non-runtime observable [4]. Common runtime observable attributes are performance, availability, security, and usability. Modifiability, integrability, reusability, portability, and testability quality attributes are common non-runtime observable. To account for system's behavior, there exist also the functionality quality attribute. The collection of runtime and non-runtime quality attributes serve as a basis toward specifying alternative architectures where each architecture is an instantiation of quality attributes tradeoffs. The tradeoffs capture interaction among quality attributes and their ranking with an underlying assumption that there can be no architectural specification that can satisfy all quality attributes[4].

To exemplify the interplay of architectural style and quality attributes, we consider a university library and its on-line system where patrons have access to hardcopy materials such as books and archival periodicals as well as softcopy materials and other on-line information via a web portal. Suppose we are asked to recommend an architectural style[5] satisfying availability and portability quality attributes. One recommendation may suggest the base architectural style to be a layered and transactional database-repository with call-and-return and client-server architecture flavor (see Table 2). The layered style provides portability for the client-side and if necessary for the server-side. To support high degree of availability (e.g., > 99.9%), we can use multiple servers either in centralized or decentralized fashion. The centralized approach suggests having two or more identical servers in one physical location where the system can automatically switch from one server to another as necessary (replace data storage unit.). In a decentralized setting, servers are dispersed in different locations thus ensuring availability – for example network (e.g., router/switch) or power failure. Client/server fits the general pattern of use of services offered by university library on-line system and dispersed patrons. It supports the main scheme of many users accessing the on-line system. Data-centered style is useful and important for satisfying the functionality requirement. The server-side data (e.g., library holdings) needs to be stored in a centralized database for frequent access by large number of patrons.

Architectural Styles	Control Issues			Data Issues		
	Topology	Synchronicity	Binding Time	Topology	Synchronicity	Mode
Call-based client/server	star	synchronous	wt, ct, rt	star	spor., lvol.	passed
Layered	hierarchical	any	wt, ct, it, rt	hierarchical	spor., lvol., cont.	any
Decentralized servers	arbitrary	asynchronous	wt, ct, rt	arbitrary	spor., lvol.	passed

Table 1: Candidate Architectural Style for University Library On-line Example

It is important to note the data/control issues as they show how static and dynamic aspects of each architectural style relates to satisfying some desired quality attributes. The control topology is principally of star pattern between library patrons and the library on-line system. Given the layered architectural style, control topology is hierarchical for a single server and arbitrary if multiple servers are used (set of decentralized servers). Control is asynchronous since there is no need to synchronize users – users interactions with the system are independent of one another. The overall data topology is arbitrary and hierarchical with multiple servers. Data continuity is also frequently sporadic and low volume (small amount of data is transmitted from client to server). Binding time is of all types – wt(write-time),

[4] There are three complementary higher level categories of quality attributes referred to as system, architecture, and business [4].
[5] It is important to note that the above recommendation is subject to business quality attributes – use of cost, market, tools, etc. may result in other architectural styles!

ct(compile-time), it(invocation-time) and rt(run-time). Data to and from clients are primarily "passed" although the database content is "shared". Table 1 shows typical settings for data and control for the architectural styles [4].

3 Intelligent Systems Lifecycle Architecture

Given the foregoing discussions, we can consider the role of architectural style as the first step toward specifying the architecture of an intelligent system. Before we proceed further, we briefly discuss performance and intelligence attributes which are considered as necessary for any intelligent system[6].

3.1 Performance Attribute

Performance attribute represents how well (response time) a system responds to stimulus both internally and externally (e.g., a robot decision to abandon a subgoal and pursue another; heed a warning send by a remote sensor). An architecture specification can satisfy performance quality by controlling the communications among components. For example, the number of interactions and the size and number of data/control can be minimized to increase performance. Of course, performance also depends on how fast each component can carry out its responsibilities, interdependencies of a component's processing on other components, hardware components (e.g., processing speed of a router and network bandwidth) and architectural topology (e.g., physical separation between computing nodes), etc. The key point is that the architectural style is the first design artifact which can be used to specify and assess performance attribute in a formal setting. Furthermore, since architectural style design decisions feed into the software and system architecture, we are able to track it from its abstract to its concrete specifications. This ability has a profound consequence – i.e., formally trace the satisfiability of performance attribute from its inception to its realization within the architecture as shown in Figure 1.

3.2 Intelligence Attribute

Now, we introduce *intelligence* as one additional quality attribute to those given in [4]. Treating intelligence as a quality attribute is based on the concept that it is distinct from the *functionality* quality attribute in the sense that a system may produce some desired behavior without necessarily using its intelligence faculty[7]. Consider the definition of intelligence as proposed by [7] – "intelligence is the ability of a system to act appropriately in an uncertain environment, where an appropriate action is that which increases the probability of success, and success in the achievement of behavioral subgoals that support the system's ultimate goal". This definition suggests, among other things, the notion that an intelligent system may achieve its goal but not necessarily with the same success rate as compared with one that is considered intelligent. Thus, we suggest the intelligence quality attribute to be distinct from the other quality attributes. Clearly, intelligence considered as a quality attribute implies that it plays a key role in the architecture lifecycle – i.e., some architectural styles may provide for intelligence while others may not and therefore some software/system architecture may or may not be categorized as "intelligent" by examining its intelligence quality attribute. Similar to other quality attributes, intelligence quality attribute interacts with other quality attributes (e.g., intelligence and security quality attributes may interact with another).

3.3 Use of Quality Attributes

As we have discussed, software/system architectures increasingly play a central role in the development of large-scale complex intelligent systems. This is primarily due to the fact that architectural specifications collectively contain the necessary knowledge required for describing a system's performance, availability, testability, functionality, etc. Intuitively, performance and intelligence of an intelligent system should therefore be traceable to its architecture. For example, for an semi-autonomous vehicle to be able to restructure itself in pursuit of achieving its objectives, it may need to account for quality attributes such as intelligence. The architecture of a semi-autonomous vehicle navigating itself to its destination through an unknown environment must account for the intelligence and performance quality attributes. Concrete realization of these quality attributes may be arriving at the destination within 5% of a specified time period with high-degree of intelligent decision-making to avoid collision – e.g., 98% collision avoidance rate.

---

[6] Reference to performance and intelligence as "attributes" may not necessarily be in agreement with the characterizations given in [6] and [7].
[7] Similar arguments can be given for other quality attributes.

This view suggests *abstract* and *concrete* quality attributes where architectural style accounts for the former and software/system architecture accounts for the latter. Given abstract and concrete quality attributes, we may ask the following questions in order to substantiate the premise that intelligent systems can be specified with the architecture lifecycle using quality attributes (see Figure 1).

1. do abstract quality attributes form a primitive basis for specifying the architecture of intelligent systems?
2. can abstract quality attributes be evolved to their concrete counterparts through the architecture specification lifecycle?

We answer our questions informally. An affirmative response to the first question implies that architectural style is the most suitable specification to capture quality attributes. Architectural style by definition is void of domain-specific knowledge. Furthermore, architectural style is sufficiently rich to account for quality attributes – its structure, components, and interactions can account for quality attributes in an abstract setting – independent of explicit knowledge for a system under consideration. Consider, a layered architectural style. This style supports dividing tasks into those aimed for a specific application and those generic to many applications. This accounts for the hierarchical decision-making. The layering approach also allows lower levels to be specialized for distinct computing platforms and thus supports the portability quality attribute. Intelligence and portability attributes are possible without domain specific knowledge for a specific application. The satisfiability of these attributes can be shown by examining the constituent parts, control issues, and data issues of the architectural style[8] (refer to Table 2). For example, hierarchical data and control topologies play major roles for satisfying the modifiability and portability attributes – i.e., the control and data issues are of direct importance to allow components interact via their connectors in an abstract setting.

Constituents Parts		Control Issues			Data Issues		
Components	Connectors	Topology	Synchronicity	Binding time	Topology	Synchronicity	Mode
various	various	hierarchical	any	wt, ct, it, rt	hierarchical	spor., lvol., cont.	any

Table 2: Layered Architectural Style Specification

With an affirmative response to the first question, we now turn to the second question. The answer to this question is also affirmative but perhaps less evident. In the architectural lifecycle, reference model accounts for known (empirically proven) ways of partitioning a class of systems (e.g., Model-View-Controller) into components and interactions using a set of unit operations [4] such as abstraction, compression, and part-whole and is-a decompositions. These unit operations, of course, underlie the quality attributes – for example, the degree to which performance and availability quality attributes can be supported by the reference model. Furthermore, unit operations support reasoning about architectural styles and thus allowing measurements in qualitative/quantitative forms. This suggests that reference model serves as a vehicle to transform abstract quality attributes to those that are grounded in a particular class of systems. Therefore, the combination of the reference model and architectural style (i.e., reference architecture) serve as the intermediary for deriving the concrete quality attributes for a software/system architecture.

Quality attributes as defined within the architecture lifecycle can be used as a means to formalize varying levels of intelligence from the architectural specification point of view. For example, intelligence can be said to range from selection rules (most primitive) to synthesis of paradigm (most advanced) [7]. The intermediary levels of intelligence are combinations of selection rules, new rules, grouping of rules, synthesis of states, and synthesis of context. That is, the ability to begin with abstract quality attributes and arrive at their concrete counterparts can serve a means to formalize increasingly higher levels of intelligence.

4 Other Approaches for Developing Intelligent System Architectures

A large body of research has been devoted to the study of intelligence and intelligent systems. Many algorithms, techniques, and methodologies have been developed for intelligent systems (e.g., [7, 10-13]). Some of these have

[8] Control/data interaction for the layered architectural style is specified as having often isomorphic shapes with flow directions either being the same or opposite [4].

become the foundations for system architectures such as NIST-RCS [2], subsumption [3, 5], InteRRAP [14]. Some architectures (e.g., subsumption) emphasize lower-level information processing and evaluation (e.g., obstacle detection and motion control) while others (e.g., InteRRAP) focus on higher-level knowledge processing and decision-making (e.g., planning and model-based prediction). The latest NIST-RCS reference architecture described by Meystel and Albus [7] account for both low- and high-level processing and decision-making. They suggest sensors, actuators, sensory processing, behavior generation, world model, and value judgment to account for the RCS class of reference architectures.

5 Conclusions

We have presented an architecture lifecycle for the design of intelligent systems. We described the importance of architectural styles as the basis toward specification and development of software and architectures. We showed the key role quality attributes play and their importance in specifying and measuring an intelligent system's performance, intelligence, modifiability, etc.. An important aspect of the quality attributes is how they evolve from their abstract form (captured in an architectural style) to their concrete realizations (represented in a system architecture). Our proposition rests on the premise that an incremental and evolutionary architectural specification lifecycle can lend itself best for selecting, specifying, developing, and measuring quality measures and ultimately system architecture. Furthermore, with this approach, tradeoffs among quality attributes can be systematically accounted for and analyzed.

References

1. Albus, J.S., M. H.G., and L. R., *NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*. 1987, Robot Systems Division, Center for Manufacturing Engineering, National Technical Information Service: Gaithersburg, MD.
2. Albus, J.S., *A Reference Model Architecture for Intelligent Systems Design*. 1994, NIST: Gaithersburg, MD.
3. Brooks, R.A., *Cambrian Intelligence: the early history of the New AI*. 1999: MIT Press.
4. Bass, L., P. Clements, and R. Kazman, *Software Architecture in Practice*. The SEI Series in Software Engineering. 1998: Addison Wesley.
5. Brooks, R.A., *A Robust Layered Control System for A Mobile Agent*. IEEE Journal of Robotics and Automation, 1986. **2**(1): p. 14-23.
6. Meystel, A.M. and E.R. Messina. *Measuring the Performance and Intelligence of Systems*. *Proceedings of PerMIS*. 2000. Gaithersburg, MD: NIST.
7. Meystel, A.M. and J.S. Albus, *Intelligent Systems Architecture, Design, and Control*. 2001: John Wiley & Sons, Inc.
8. Müller, J.P. and M. Pischel, *An Architecture for Dynamically Interacting Agents*. International Journal of Intelligent and Cooperative Information Systems, 1994. **3**(1): p. 25-45.
9. Sarjoughian, H.S., B.P. Zeigler, and S.B. Hall, *A Layered Architecture for Agent-based System Development*. IEEE Proceeding, 2001. **89**(2): p. 201-213.
10. Genesereth, M., N.J., Nilsson, *Logical Foundation of Artificial Intelligence*. 1987, San Mateo, CA: Morgan Kufmann. xviii+405.
11. Saridis, G.N., *Intelligent Robotic Control*. IEEE Trans. Auto. Cont., 1983. **28**(5): p. 547-557.
12. Zadeh, L.A., *Fuzzy Logic, Neural Networks and Soft Computing*. Comm. of ACM, 1994. **37**(3): p. 77-84.
13. Zeigler, B.P., *Object-Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems*. 1990, New York: Academic Press.
14. Müller, J.P., *The Design of Intelligent Agents: A Layered Approach*. Lecture Notes in Artificial Intelligence. 1996: Springer.

Uncertain Predictions of Flow and Transport in Random Porous Media: The Implications for Process Planning and Control

Shlomo Orr, MRDS, Inc.

Traditional predictions of flow and transport in porous media are based on mass balance equations in the form of partial differential equations (PDEs), where the flux at every point is defined by Darcy's law, $\mathbf{q} = -K\nabla h$, i.e., the flux is proportional to hydraulic head gradient, where K is the hydraulic conductivity of the medium (a tensor or a scalar; essentially, a material property); it is further assumed that Darcy's law applies to transient multiphase flow in three dimensions [14,26,36,55]. The solutions of these PDEs constitute groundwater models, oil reservoir simulators, geothermal models, and models of flow and transport in soils/vadose-zone. Due to the similarity between the linear Darcy's law and Ohm's law in electricity, Fourier law in heat conduction, and Hooke's law in elasticity, such models (or PDE solutions) are similar and commonly interchangeable between these fields.

Natural porous formations are heterogeneous, and display spatial variability of their geometric and hydraulic properties. This variability is of irregular and complex nature. It generally defies a precise quantitative description because of insufficient information on all relevant scales [9,18,26, 29,30,32,33,86,91]. In practice, only sparse measurements are available (limited by cost of drilling and monitoring). Under lack of exhaustive information, the higher the variability, the higher is the uncertainty. Geostatistics is commonly used to analyze and interpolate between measurements in mining and oil explorations, as well as hydrology and soil sciences, using methods such as "kriging", where the uncertainties in "krigged" values are also quantified [33,35,36,42,56-58,76,77,81,83,84,87, 104,105]. Frequently, these data are collected on different scales that may differ from the required scale of predictions. The task of quantitatively relating measurements and properties on different scales is difficult and intriguing [4,5,7,13,27,29, 30,38-40,46,59,78,86,91,101,108,109]. Lack of information in both observed results (output) and measured material properties (parameters) causes uncertain predictions. Spatial variability and uncertainty have lead engineers and geologists to use probabilistic theories that translate the uncertainty to a random space function (RSF) or a random field, consisting of an ensemble of (infinite number of) equally probable "realizations" of parameter values, all having the same spatial statistics, particularly correlation structure [107,76,77,23,33,35,36,42,56,58,83,84, 85,87,91]. Imbedded in this approach is a geostatistical model of an assumed joint *pdf.* In practice, only the first two moments are considered, with an underlying assumption of multivariate normal distribution; in particular, the theoretical semi-variogram (or simply, "variogram") - the reciprocal of the covariance function, and the mean and variance of the *pdf.* Since these joint moments are inferred from spatial data, the assumption of ergodicity (i.e., assuming that the ensemble and spatial statistics are identical – a theorem that cannot be proven on real data) must be invoked, which, in turn, implies some kind of stationarity (or statistical homogeneity) [33,35,36,49]. Further, in order to determine the variogram model from available spatial data, an inverse method has to be used to estimate the parameters of this variogram; sensitivity to data errors on one hand, and identifiability problems (of model parameters) on the other hand [81,87,104,105] lead to uncertainty in the geostatistical model itself, which is usually ignored (in fact, the common practice is to fit the variogram model to the experimental variogram by eye and by subjective judgement of model type, degree of stationarity (drift), and statistical anisotropy). Another ignored uncertainty is in the "measured" hydraulic conductivity value that are actually inferred from hydraulic tests interpreted by simplistic models that assume local homogeneity, which is somewhat inconsistent with the RSF approach.

The use of RSF to predict behavior of uncertain systems is not limited to flow in porous media; great efforts have been devoted in all science and engineering fields to (a) estimate or predict mean behavior of the system under different stresses, and (b) compute the uncertainty associated with these predictions (expressed by the variance-covariance of the solution), i.e., the first two statistical moments of system output [4,5,7,10-12,15,16-22,25,28,31-33-35,44,47-51, 57,62,63,68-71,73-78,82-93,97,99-101,103,106, 107,110,111,112,114]. The resulted approximate solutions are usually limited to simple geometry and boundary conditions, and to moderate to low variability, as they mostly rely on variations of small-perturbation methods. Recently developed approximations for higher variability in material properties using integro-differential representations have been limited to simple 2D geometry and simple boundary conditions [91,114], with little use, yet.

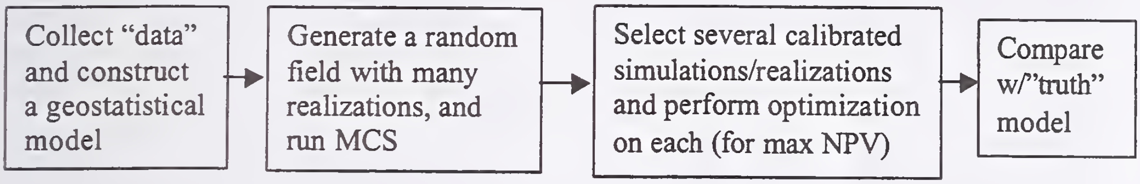
With respect to mean behavior, it is especially desired to define effective properties of heterogeneous media [1-3,7,8,25,33,37-41,43,45, 52,54,60,61,64-66,71-73,75,77,78,91,93,97,98, 102,108,109,114]. With the two statistical moments of system output, one hopes to optimize and control systems such as oil production, groundwater remediation, irrigation, leaching, etc. However, approximating the statistical behavior of a complex system of flow in random porous media based on the statistics of the hydraulic parameters is a formidable task, at best [46,114, 91-93], because this implies solving the stochastic flow and transport equations (analogous to the Heat/Diffusion Equation) or other stochastic PDEs in other fields (*ibid*). Since a direct, explicit (closed-formed) solution to the problem of random parameters (or coefficients) is practically impossible[1] [74,93,112,114], only approximate solutions have been reported in the literature for relatively simple cases [*ibid*,28,31-33,49-50,83-93,100-101]. Interest in this class of stochastic differential equations has its origins in quantum mechanics, wave propagation, turbulence theory, random eigenvalues, and functional integration [8,20,51,48,62,63,68-71,74,78,99,103,111]. Due to the limited types of problems that can be tackled by stochastic theories (closure approximations), in practice, numerical approximations in the form of high-resolution Monte Carlo simulations (MCS) are used; however, MCS require ample computer power and CPU time [46,114,88-89] Orr [114] describes and analyzes other difficulties and non-quantifiable uncertainty associated with MCS, particularly the generation of correlated random fields that are faithful to the geostatistical model, and simulations of flow in highly heterogeneous/erratic media.

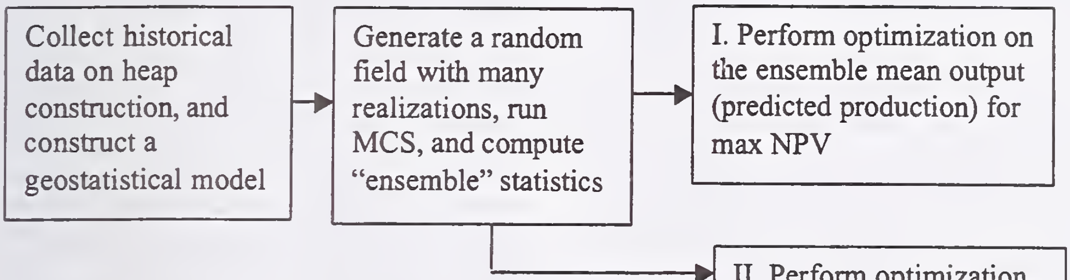
The geostatistical model provides the statistics of the parameters, particularly the permeability; in MCS, it provides the spatial distribution of parameter values for each realization. Based on these values and assumed model structure (i.e., the conceptual model of flow and transport, including large geologic features, boundary- and initial-conditions, sources and sinks), stochastic solutions are approximated (analytically or numerically). Since the model structure itself is frequently uncertain due to (a) unknown boundary and initial conditions, (b) extent of large-scale geologic features, and (c) information pertaining to geochemical reactions and phase transition, such sets of 500-1000 MCS need to be repeated for several, if not many alternative conceptual models or equally probable model structures [116]. Subsequent optimization requires many repetitions of each Monte Carlo simulation in order to build the search space (i.e., generate sufficient number of scenarios or trajectories); hence, rigorous optimization under uncertainty is prohibitive in terms of computer power and time for most practical applications. In an attempt to optimize best new well placement in an oil reservoir, Guyaguler and Horne [53] were forced to perform optimization on only 23 randomly selected, "history matched" realizations in order to overcome the obstacle of prohibitive computer power and time, while continuously verifying their results against a "truth" model (apparently based on extensive calibration and some "effective" properties). Indeed, the authors concluded that "a decision based on a single realization (though with perfect history match) may differ substantially from the true optimum"[53, p.4]. As was shown theoretically by Neuman and Orr [91], unique effective properties (of random media) that are data independent do not generally exist except for a few special cases. This explains why parameter estimates obtained by traditional inverse methods tend to vary as one modifies the database and/or the imposed stress [113,115]; consequently, calibration of deterministic models may be meaningless in term of predictive power.

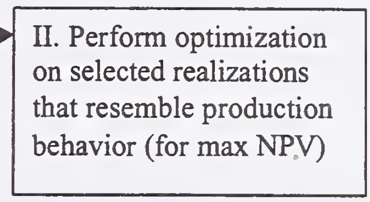
Thus, on the way to optimal solutions using stochastic predictions we already encounter

[1] A solution to a stochastic PDE consists of specifying the (joint) probability density function (*pdf*) of the response, h(x), given those of K(x) (and forcing functions and boundary conditions). Unfortunately, one cannot obtain the joint cumulative distribution function (CDF) of the random response at **all** (infinite number of) points. Even for a finite set of points, one cannot obtain closed-form equations for a finite number of moments. This problem can be circumvented by either approximations (e.g., perturbation methods, Neumann series) or by numerical approximations, i.e., Monte Carlo simulations (MCS).

significant theoretical and practical barriers, particularly, uncertain interpretations, model limitations, prohibitive computer power, and non-existing effective parameters – all of which render these predictions highly uncertain, while only part of this uncertainty is being quantified. Our presentation will discuss these barriers, and will bring two examples where traditional stochastic approximations (i.e., approximate solutions of the governing stochastic PDEs) are unable to provide reliable practical solutions. One example deals with the difficult problem of finding the best location of the next well in an oil reservoir [53].). The following simplified block diagram describes the work flow as described by Guyaguler and Horne [53]. Note that "data" are inferred values from field tests. Note also that at each step, there are inherent errors, inconsistencies, and unaccounted (as well as counted) uncertainty. Another example involves optimal control of heap leaching in the mining industry [94,95].







In both cases:

1. Rigorous MCS is already prohibitive in terms of computing resources
2. Models cannot capture the full complexity; hence, predictions are unreliable
3. Rigorous optimization is impossible due to time and computer limitations

In each of these cases, decisions have been made at every step of the solution. Many of these decisions are made subjectively, based on experience, knowledge, thoroughness, understanding (or conceptual models of the process), computer resources, and time limitation, possessed by the modeler. These factors affect and are being affected by the degree of belief (by the modelers) in each decision made.

Initially, sampling (network) design decisions have to be made re sampling locations (an optimization procedure on its own that depends on the end results as well, i.e., a feedback mechanism with the goal of minimizing prediction uncertainty [119]). Then, the following decisions (and sub-decisions) have to be made: (a) type of model and/or curve fitting to use to determine the hydraulic conductivity or permeability (a mini-inverse model that could be ill-posed); in the case of two-phase flow (e.g., oil-water in a reservoir, water-air in a heap), several other decisions (or assumptions) have to be made, and an ill-posed inverse procedure must take place [117-121, and author's personal experience]; (b) determining and eliminating outliers; (c) determining optimal lag distance for the experimental variogram; (d) determining the *pdf,* variogram model, and model parameters; particularly, choosing between Gaussian and Indicator models, judging between drift and/or anisotropy, and determining the drift (requires a complex inverse procedure, with typical ill-posed cases; see [36,81,104,105]); (e) determining dimensionality, domain size, and mesh resolution with respect to correlation scales, measurement scales, and property/parameter representation scales; (f) determining conceptual model (or model structure) of flow and transport, including the parameters to be treated as random, the governing equations (PDEs, including reactive transport and multiphase flow), and uncertain boundary- and initial-conditions, boundary locations, zones of specific character/features (based on geologic and hydraulic information) – i.e., the simulator (these decisions should be made first, and re-evaluated as more information is being analyzed); (g) for MCS: random number generator (RNG; portable or not; RNG type; seed, etc.); (h) random field generator (RFG), including type and RFG parameters (in addition to the variogram model and *pdf*); (i) number of simulations (should be based on the behavior of point output variance as a function of the number of simulations, i.e., a trial & error procedure which usually is not being done, including in the above two cases), given the limited computer power); (j) number of realizations required for optimization; ideally, all (hundreds of) realizations are used; one compromise may be reducing the number of realizations [53]; another compromise would consider mean system behavior (using the resulting ensemble mean results, and optimize that mean behavior (a major uncertain decision); (k) type of optimization/search algorithms (l) number of simulations (per realization) for constructing a sufficiently dense search state-space; (m) objective function and cost variables. The last three decisions have to be made by all optimization procedures.

In the well placement problem [53], due to limited computer power, a decision was made to use only 23 realizations (while 500-1000 simulations are typically needed to provide meaningful ensemble statistics [114,46]), and a decision was made to calibrate randomly selected random realizations, individually, which contradicts the concept of random fields (or RSF), but may have served a practical purpose (i.e., to find the maximum NPV). Similar prohibitive computer power problem prevails in the heap leaching simulations. While unstable oil-water fronts in the well placement case cannot be captured by the simulator, unstable wetting fronts during heap leaching cannot be captured even by high-resolution two-phase models (like the one used by Orr and Vesselinov [95]). In the latter case, lack of information on essential reactive transport properties, and unmonitored dynamic changes in heap structure (particularly sealing of pore space by clays and erosion products) cannot be determined and modeled. Consequently, the simulators are weak, missing on mean system behavior (or predictions) with unaccounted uncertainty, resulting in weak optimization and control.

We see that along the track of approximate solutions and partial optimization of oil reservoirs and heap leaching operations based on predetermined stochastic PDEs, the confidence of modelers and decision-makers is being eroded with each decision being made, depending on their knowledge and degree of belief at each decision point. By the end of this process, decision-makers find themselves with very little confidence and very little decision power. Commonly, in this stochastic approach, the degree of belief is not quantified, though it contributes to the total uncertainty. Alternative fuzzy logic techniques do quantify the uncertainty associated with the degree of belief.

We therefore propose to replace these formidable stochastic approaches by a simpler yet intelligent stochastic control, particularly, the multiresolution decision support system (MRDS, which includes fuzzy logic as one of its components) in order to reach more reliable and efficient optimal solutions, with reduced, accountable uncertainty, in real time, with minimal computer resources. Moreover, unlike the rigid stochastic PDEs, MRDS can be naturally extended to optimal control of linked processes. In the oil field case, this includes exploration, all surface installations and operations, delivery system, and distribution. In the heap leaching case, this includes subsequent solvent extraction and electrowinning, as well as all antecedent processes – from exploration and blasting to transportation, crushing, agglomeration, conveying, placement, and design of the irrigation systems.

REFERENCES

1. Ababou, R., Identification of effective conductivity tensor in randomly heterogeneous and stratified aquifers, *5th Canadian-American Conference on Hydrogeology*, Sept. 18-20, Calgary, Alberta, Canada, 1990.
2. Ababou, R. and E.F. Wood, Comment on ``Effective groundwater model parameter values: Influence of spatial variability of hydraulic conductivity, leakance, and recharge" by J.J. Gomez-Hernandez and S.M. Gorelick, *Water Resour. Res.* } 26(8), 1843-1846, 1990a.
3. Ababou, R. and L.W. Gelhar, Self-similar randomness and spectral conditioning: analysis of scale effects in subsurface hydrology, *Dynamics of fluids in hierarchical porous media*, edited by J.H. Cushman, Academic Press, 393-428, 1990.
4. Adomian, G., *Stochastic Systems*, Academic Press, New York, 1983.
5. Ahmed, N.U., *Semigroup Theory with Applications to Systems and Control*, Longman Scientific & Technical with John Wiley & Sons, New York, 1991.
6. Albus, J., A. Meystel, S. Uzzaman, "Nested Motion Planning for an Autonomous Robot", Proc. IEEE Conference on Aerospace Systems, May 25-27, Westlake Village, CA, 1993

7. Auriault, J.L., Heterogeneous medium. Is an equivalent macroscopic description possible? *Int. J. Engng. sci.* 29(7), 785-795, 1991.
8. Avellanda M., and A.J. Majda, Mathematical models with exact renormalization for turbulent transport, *Comm. Math. Phys.*, 131, 381-429, 1990.
9. Bachu, S., and D. Cuthiell, Effects of core-scale heterogeneity on steady-state and transient fluid flow in porous media: Numerical analysis, *Water Resour. Res.* 26(5), 863-874, 1990.
10. Bakr, A.A., Effect of Spatial Variations of Hydraulic Conductivity on Groundwater Flow, Ph.D. dissertation, N.M. Inst. of Min. and Technol., Socorro, 1976.
11. Bakr, A.A., L.W. Gelhar, A.L. Gutjahr, and J.R. MacMillan, Stochastic analysis of spatial variability in subsurface flows 1. Comparison of one- and three dimensional flows, *Water Resour. Res.*, 14(2), 263-271, 1978.
12. Batchelor, G.K., Transport properties of two phase materials with random structure, *Annual Reviews of Fluid Mechanics*, 6, 227-254, 1974.
13. Baveye, P. and G. Sposito, Macroscopic balance equations in soils and aquifers: The case of space- and time-dependent instrumental response, *Water Resour. Res.*, 21(8), 1116-1120, 1985.
14. Bear, J., *Dynamics of Fluids in Porous Media*, American Elsevier, 1972.
15. Bellman, R.E., Stochastic transformations and functional equations. In *Stochastic Processes in Mathematical Physics and Engineering*, edited by R.E. Bellman, Vol.XVI, 171-177, Amer. Math. Soc., 1963.
16. Benaroya, H. and Rehak, M., The Neumann series/born approximation applied to parametrically excited stochastic systems. *Probabilistic Engineering Mechanics*, 2(2), 1987.
17. Benaroya, H. and Rehak, M., Finite element methods in probabilistic structural analysis: A selective review, *Applied Mechanics R*eviews, 41(5), 201-213, 1988
18. Beran, M.J., *Statistical Continuum Theories*, Interscience, New York, 1968.
19. Bharucha-Reid, A.T., *Random Integral Equations*, Mathematics in Science and Engineering, Vol.96, Academic Press, 1972.
20. Brenner, H., A general theory of Taylor dispersion phenomena, II, An extension. *PCH, Physicochem. Hydrodyn.*, 3(2), 139-157, 1982.
21. Cheng, A.H.-D. and O.E. Lafe, Boundary element solution for stochastic groundwater flow: Random boundary condition and recharge, *Water Resour. Res.*, 27(2), 231-242, 1991.
22. Christakos, G., *Random Field Models in Earth Sciences*, Academic Press, 1992.
23. Clifton, P.M. and S.P. Neuman, Effects of Kriging and inverse modeling on conditional simulation of the Avra Valley aquifer in Southern Arizona, *Water Resour. Res.*, 18(4), 1215-1234, 1982.
24. Corson, C., A. Meystel, F. Otsu, S. Uzzaman, "Semiotic Multiresolutional Analysis of a Power Plant", in Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems, Proc. of the 1995 ISIC Workshop, Monterey, CA, pp. 401-405
25. Cushman, J. H., Volume averaging, probabilistic averaging, and ergodicity, *Adv. Water Resour.*, 6, 182-184, 1983.
26. Cushman, J.H., On unifying the concepts of scale, instrumentation, and stochastics in the development of multiphase transport theory, *Water Resour. Res.*, 20(11), 1668-1676, 1984.
27. Cushman, J.H., On measurement, scale, and scaling, *Water Resour. Res.*, 22(2), 129-134, 1986.
28. Cushman, J.H., Development of stochastic partial differential equations for subsurface hydrology, *Stochastic Hydrology and Hydraulics*, 1(4), 241-262, 1987.
29. Cushman, J.H., An introduction to hierarchial porous media, *Dynamics of fluids in hierarchical porous media*, edited by J.H. Cushman, Academic Press, 1-5, 1990.
30. Cushman, J.H., On the inseparability of the measurement process from theories of transport in heterogeneous materials, in *Concise Encyclopedia of Environmental Systems*, edited by P. Young, in press, 1991.
31. Dagan, G., Models of groundwater flow in statistically homogeneous porous formations, *Water Resour. Res.*, 15(1), 47-63, 1979.
32. Dagan, G., Statistical theory of groundwater flow and transport: Pore to laboratory, laboratory to formation, and formation to regional scale, *Water Resour. Res.*, 22(9), 120S-134S, 1986.
33. Dagan, G., *Flow and Transport in Porous Formations*, Springer-Verlag, New York, 1989.
34. Dederichs, P.H., and R. Zeller, Variational treatment of elastic constants of disordered materials, *Z. Phys.*, 259, 103-116, 1973.
35. Delhomme, J.P., Spatial variability and uncertainty in groundwater flow parameters: A geostatistical approach, *Water Resour. Res.*, 15(2), 269-280, 1979.
36. de Marsily, G., *Quantitative Hydrogeology*, Academic Press, 1986.
37. Desbarats, A.J., Numerical estimation of effective permeability in sand-shale formations, *Water Resour. Res.*, 23(2), 273-286, 1987.
38. Desbarats, A.J., Spatial averaging of hydraulic conductivity in three-dimensional heterogeneous porous media, *Math. Geol*}, Vol. 24(3), 249-267, 1992a.
39. Desbarats, A.J., Spatial averaging of transmissivity in heterogeneous fields with flow toward a well, *Water Resour. Res.*, 28(3), 757-767, 1992b.
40. Desbarats, A.J. and R. Dimitrakopoulos, Geostatistical modeling of transmissibility for 2D reservoir studies, *SPE Formation Evaluation*, 5(4), 437-443, 1990.
41. Deutsch, C., Calculating effective absolute permeability in sandstone/shale sequences, *SPE Form. Eval.*, Sept. 343-348, 1989.
42. Deutsch, C.V., and A.G. Journel, *GSLIB: Geostatistical Software Library and User Guide*, Oxford University Press, 1992.
43. Dykaar, B.B. and P.K. Kitanidis, Determination of the effective hydraulic conductivity for heterogenous porous media using a numerical spectral approach 1. Method, *Water Resour. Res.*, 28(4), 1155-1166, 1992a.
44. Elishakoff, L., *Probabilistic Methods in the Theory of Structures*, John Wiley \& Sons, New York, 1983.
45. El-Kadi, A.I. and W. Brutsaert, Applicability of effective parameters for unsteady flow in nonuniform aquifers, *Water Resour. Res.*, 21(2), 183-198, 1985.
46. Follin, S., Numerical Calculations on Heterogeneity of Groundwater Flow, Ph.D. dissertation, The Royal Inst. of Technol., Stockholm, 1992.
47. Freeze, R.A., A stochastic-conceptual analysis of one-dimensional groundwater flow in nonuniform homogeneous media, *Water Resour. Res.*, 11(5), 725-741, 1975.
48. Frisch, U., Wave propagation in random media, in *Probabilistic Methods in Applied Mathematics*, edited by A.T. Bharucha-Reid, 1, 75-198, Academic Press, New York, 1968.
49. Gelhar, L.W., Stochastic subsurface hydrology. From theory to applications, *Water Resour. Res.*, 22(9),

135S-145S, 1986.

50. Gelhar, L. W., Stochastic analysis of solute transport in saturated and unsaturated porous media, in *Advances in Transport Phenomena in Porous Media*, NATO ASI Ser., edited by J. Bear and M. Y. Corapcioglu, 657-700, Martinus Nijhoff, Dordrecht, Netherlands, 1987.
51. Ghanem, R.G., and P.D. Spanos, *Stochastic Finite Elements: A spectral Approach*. Springer-Verlag, New York, 1991.
52. Gomez-Hernandez, J.J. and S.M. Gorelick, Effective groundwater model parameter values: Influence of spatial variability of hydraulic conductivity, leakance, and recharge, *Water Resour. Res.*, 25(3), 405-419, 1989.
53. Guyaguler, B. and R.N. Horne, Uncertainty Assessment of Well Placement Optimization, *SPE 71625*, 2001.
54. Hashin, Z., and S. Shtrikman, A variational approach to the theory of effective magnetic permeability of multiphase materials, *Jour. Of Appl. Physics*, 33, 3125-3131, 1962.
55. Hassanizadeh, M. and W.G. Gray, General conservation equations for multi-phase systems, 1, Averaging procedure, *Adv. Water Resour.*, 2, 131-144, 1979.
56. Isaaks, E.H. and R.M. Srivastava, *An Introduction to Applied Geostatistics*}, Oxford University Press, 1989.
57. Johnson, M.E., *Multivariate Statistical Simulation*, John Wiley & Sons, New York, 1987.
58. Journel, A.G. and Ch.J. Huijbregts, *Mining Geostatistics*, Academic Press, 1978.
59. Kadanoff, L.P., Scaling and Universality in statistical physics, *Phisica A*, 163, 1-14, 1990.
60. Kasap, E., and L.W. Lake, An analytical method to calculate the effective permeability tensor of a grid block and its application in an outcrop study, *paper presented at the SPE Symp. on Reser. Sim.*, Soc. of Pet. Eng. Houston, 1989.
61. Kasap, E., and L.W. Lake, Calculating the effective permeability tensor of a grid block, *SPE Form. Eval.*, June, 192-200, 1990.
62. Keller, J.B., Wave propagation in random media, *Proc. Symp. Appl. Math.*, 13, 227-246, Amer. Math. Soc., Providence, Rhode Island, 1962.
63. Keller, J.B., Stochastic equations and wave propagation in random media, *Proc. Symp. Appl. Math.*, 16, 145-170, Amer. Math. Soc., Providence, Rhode Island, 1964.
64. Keller, J.B., Effective conductivity of reciprocal media, in *Random Media*, edited by G. Papanicolaou, The IMA Vol. in Math. and its Appl., Vol.7, Springer-Verlag, New York, 1987.
65. King, P.R., The use of renormalization for calculating effective permeability, *Transport in Porous Media*, 4, 37-58, 1989.
66. Kitanidis, P.K., Effective hydraulic conductivity for gradually varying flow, *Water Resour. Res.*, 26(6), 1197-1208, 1990.
67. Kohler, W.E., and G. Papanicolaou, Bounds for the effective conductivity of random media, in *Macroscopic properties of Disordered Media*, edited by R. Burridge, S. Childress, and G. Papanicolaou, *Lecture Notes in Physics*, 154, Springer-Verlag, New York, 1982. 111-130
68. Kraichnan, R.H., Dynamics of nonlinear stochastic systems, *J. Mathematical Phys.*, 2, 124-148, 1961.
69. Kraichnan, R.H., The closure problem of turbulence theory, *Proc. Symp. Appl. Math.*, 13, 199-225, Amer. Math. Soc., Providence, Rhode Island, 1962.
70. Kraichnan, R.H., Eddy viscosity and diffusivity: exact formulas and approximations, *Complex Systems*, 1, 805-820, 1987.
71. Kröner, E., Bounds on effective elastic moduli of disordered materials, J. *Mech. Phys. Solids*, 25, 137-155, 1977.
72. Landau, L.D. and E.M. Lifshitz, *Electrodynamics of Continuous Media*, Pergamon Press, Oxford, England, 1960.
73. Landauer, R., Electrical conductivity in inhomogeneous media, in *Electrical Transport and Optical Properties of Inhomogeneous Media*, edited by J.C. Garland and D.B. Tanner, AIP Conference Proceedings, No.~40, American Institute of Physics, 1978.
74. Lumely, J., and A. Panofski, *The Structure of Atmospheric Turbulence*. John Wiley \& Sons, New York, 1964.
75. Markov, K.Z., Application of Volterra-Wiener series for bounding the overall conductivity of heterogeneous media. I. General procedure, *SIAM J. of Appl. Math.*, 4, 831-849.
76. Matheron, G., The intrinsic random functions and their applications, *Advan. Appl. Prob.*, 5, 439-468, 1973.
77. Matheron, G., *The Theory of Regionalized Variables and its Applications*, Les cahiers du contre de morphologie mathematique de Fontainbleau, No.5, 1971.
78. Mei, C.C. and J.-L. Auriault, Mechanics of heterogeneous porous media with several spatial scales, *Proc. R. Soc. London, Ser. A*, 426, 391-423, 1989.
79. Meystel, A. and S. Uzzaman, "Multiresolutional Decision Support System", US Patent, No. 6102958 of August 2000
80. Meystel, A. and J. Albus, *Intelligent Systems: Architectures, Design, Control*, Wiley, NY 2001
81. Myers, D.E., A series of lecture notes on conditional simulations, in *Advanced Geostatistics* course, University of Arizona, 1990.
82. Nashed, M.Z., and H.W. Engle, Random generalized inverse and approximate solutions of random operator equations, in *Approximate Solutions of Random Equations*, edited by A.T. Bharucha-Reid, North Holland, New York, 1978.
83. Neuman, S.P., Statistical characterization of aquifer heterogeneities: an overview. In *Recent trends in hydrology*, edited by T.N. Narasimhan, Special Paper 189, 81-102, Geol. Soc. Am. Spec. Pap. 1982.
84. Neuman, S.P., Role of geostatistics in subsurface hydrology, in *NATO ASI Series, Geostatistics for Natural Resources Characterization*, edited by G. Verly, M. David, A.G. Journel, and A. Marechal, D. Reidel, Hingham, Mass., 2, 127-140, 1984.
85. Neuman, S.P., Stochastic continuum representation of fractured rock perrmeability as an alternative to the REV and fracture network concepts, *28th US Symposium on Rock Mechanics*, edited by I.W. Farmer, J.J.K. Daemen, C.S. Desai, C.E. Glass, and S.P. Neuman, Balkema, Rotterdam, The Netherlands, 533-561, 1987.
86. Neuman, S. P., Universal scaling of hydraulic conductivities and dispersivities in geologic media, *Water Resour. Res.*, 26(8), 1749-1758, 1990.
87. Neuman, S.P. and E.A. Jacobson, Analysis of nonintrinsic spatial variability by residual kriging with application to regional groundwater levels, *Math. Geol.*, 16(5), 499-521, 1984.
88. Neuman, S.P., S. Orr, O. Levin, and E. Paleologos, Theory and high-resolution finite element analysis of 2-D and 3-D effective permeabilities in strongly heterogeneous porous media, in *Computational Methods in Water Resources IX*, edited by T.F. Russell, R.E. Ewing, C.A. Brebbia, W.G. Gray, and G.F. Pinder, Elsevier Applied Science, New York, 1992.
89. Neuman, S.P., O. Levin, S. Orr, E. Paleologos, D. Zhang, and Y.~-K. Zhang Nonlocal representations of subsurface flow and transport by conditional moments, *Computational Stochastic Mechanics*, edited by A.H.D.

Cheng, and C.Y. Young, Computational Mechanics Publications, 1992.
90. Neuman, S.P., Eulerian-Lgrangian theory of transport in space-time nonstationary velocity fields: 1. Exact nonlocal formalism by conditional moments and weak approximation, *Water Resour. Res.*, 1993.
91. Neuman, S. P., & Orr S., Prediction of steady state flow in non uniform geologic media by conditional moments: Exact non local formalism, effective conductivities, and weak approximations, *Water. Resour. Res.*, 29 (2), 341-364, 1993.
92. Neuman, S. P., Tartakovsky, D. M., Wallstrom, T. C. & Winter, C. L., Correction to "Prediction of steady state flow in nonuniform geologic media by conditional moments: Exactnonlocal formalism, effective conductivities, and weak approximation", *Water Resour. Res.*, 32(5), 1479-1480, 1996.
93. Orr, S. and S.P. Neuman, "Operator and Integro-Differential Representations of Conditional and Unconditional Stochastic Subsurface Flow". An invited paper, Journal of Stochastic Hydrology and Hydraulics, Vol 8(2), pp. 157-172, 1994.
94. Orr, S. "Enhanced Heap Leaching - I. Insights", ME/SME Journal Vol. 54, No. 8, 2002.
95. Orr, S. and V. Vesselinov, "Enhanced Heap Leaching – II. Applications", ME/SME Journal, Vol. 54, No. 8, 2002.
96. Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1984.
97. Paleologos, E., Effective Hydraulic Conductivity of Bounded, Strongly Heterogeneous Porous Media, Ph.D. dissertation, University of Arizona, 1993.
98. Poley, A.D., Effective permeability and dispersion in locally heterogeneous aquifers, *Water Resour. Res.*, 24, 1921--1926, 1988.
99. Roberts, P.H., Analytical theory of turbulent diffusion, *Jour. Fluid. Mech.*, 11, 257-283, 1961.
100. Rubin, Y. and G. Dagan, Stochastic analysis of boundaries effects on head spatial variability in heterogeneous aquifers 1. Constant head boundary, *Water Resour. Res.*, 24(10), 1689-1697, 1988.
101. Rubin, Y., and Gomez-Hernandez, J.J., A stochastic approach to the problem of upscaling of conductivity in disordered media: theory and unconditional numerical simulations, *Water Resour. Res.*, 26(4), 691-701, 1990.
102. Saez, A.E., C.J. Otero, and I. Rusinek, The effective homogeneous behavior of heterogeneous porous media, *Transport in Porous Media*, 4, 213-238, 1989.
103. Saffman, P.G., On the boundary condition at the surface of a porous medium, *Studies in Applied Math.*, L(2), 93-101, 1971.
104. Samper, F.J. and S.P. Neuman, Estimation of spatial covariance structures by adjoint state maximum likelihood cross validation, 1. Theory, *Water Resour. Res.*, 25(3), 351-362, 1989a.
105. Samper, F.J. and S.P. Neuman, Estimation of spatial covariance structures by adjoint state maximum likelihood cross validation, 3. Application to hydrochemical and isotopic data, *Water Resour. Res.*, 25(3), 373-384, 1989b.
106. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, John Wiley \& Sons, New York, 1980.
107. Shvidler, M.I., Filtration Flows in Heterogeneous Media (in Russian), *Izv. Akad. Nauk SSSR Mech.*, 3, 185-190, 1962.
108. Tartar, L., Remarks on homogenization, *Homogenization and Effective Moduli in Materials and Media*, edited by J.L. Ericksen, D. Kinderlehrer, R. Kohn, and J.L. Lions, The IMA Vol. in Math. and its Appl., Vol.7, 228-246, Springer-Verlag, New York, 1986.
109. Tartar, L., Nonlocal effects induced by homogenization, in *Partial differential equations and the calculus of variations, Vol. 2, Essays in honor of Ennio De Giorgi*, edited by F. Colombini, A. Marino, L. Modica, and S. Spagnolo, *Progress in nonlinear differential equations and their applications*, 925-938, Birkhauser, Boston, 1989.
110. Tsokos C.P., and W.J. Padgett, *Random Integral Equations with Applications to Life Sciences and Engineering*, Mathematics in Science and Engineering, 108, Academic Press, New York, 1974.
111. Van Kampen, N. G., *Stochastic Processes in Physics and Chemistry*, North-Holland, Amsterdam, 1981.
112. Vanmarcke, E.H., *Random Fields: Analysis and Synthesis*, MIT Press, Cambridge, Mass., 1983.
113. Carrera, J., J. Heredia, S. Vomvoris, and P. Hufschmied, Fracture flow modelling: Application of automatic calibration techniques to a small fractured monzonitic gneiss block, in *Hydrogeology of Low Permeability Environments*, edited by S.P. Neuman and I. Neretnieks, Hydrogeology, Selected Papers, Intern. Assoc. Hydrogeol., Verlag Heinz Heise, Hannover, Germany, 2, 115-167, 1990.
114. Orr, S., Stochastic Approach to Steady State Flow in Nonuniform Geologic Media, Ph.D. dissertation, University of Arizona, 1993.
115. Neuman, S.P., and A. Guadagnini, A new look at traditional deterministic flow models and their calibration in the context of randomly heterogeneous media. *MODELCARE*-1999.
116. James, A.L., and C.M. Oldenburg, Linear and Monte Carlo uncertainty analysis for subsurface contaminant transport simulation. *Water Resources Research*, Vol. 33, No. 11. pp. 2595-2508. 1997.
117. Carrera, J., Neuman, S.P, Estimation of aquifer parameter under transient and steady state conditions : 1. Maximum likelihood method incorporating prior information. Water Resources Research, v. 22, 2, p. 199-210, 1986a.
118. Carrera, J., Neuman, S.P., Estimation of aquifer parameter under transient and steady state conditions : 2. Uniqueness, stability, and solution algorithms. Water Resources Research, v. 22, 2, p. 211-227, 1986b.
119. Carrera, J., Neuman, S.P., Estimation of aquifer parameter under transient and steady state conditions : 3. Application to synthetic and field data. Water Resources Research, v. 22, 2, p. 228-242, 1986c.
120. Carrera, J., State of the art of inverse problem applied to the flow and solute transport equations. Groundwater Flow and Quality Modeling, Custodio, E. et al., eds., NATA ASI Series, Serie C : Mathematical and Physical Sciences, v. 224, D. Reidel Pub. Co. Dordrecht, The Netherlands, p. 549-583, 1987.
121. Gh. de Marsily, J.P. Delhomme, A. Coudrain-Ribstein, A.M. Lavenue, Four Decades of Inverse Problems in Hydrogeology, in celebration of S.P. Neuman's 60th Birthday, 30 pp., 1998.

APPENDICES

APPENDIX A
WORKSHOP SCHEDULE

PerMIS 2002 Workshop Schedule

Tuesday, August 13, 2002

Registration starts at 7.30 A. M.

Green Auditorium

8:15-8:20 **Welcome by Workshop Chairs**
8:20-8:45 **Dale Hall, MEL Director:** ***Introduction***
8:45-9:15 **Research Problems of Performance Measuring**
A. Meystel, Drexel University and National Institute of Standards & Technology

9:15 – 9:30 – Coffee Break

9:30 – 12:00 – Morning Sessions 1M

Session 1M1 – Performance Metrics – Lecture Room A

Chairs: S. Agrawal, H. Yanco

1. ***Performance Metrics for Intelligent Systems: An Engineering Perspective***
   R. Gao, Purdue University
   L. Tsoukalas, Purdue University
2. ***RCS Based Hardware-in-the-loop Intelligent System Design & Performance Measurement***
   S. Ananthakrishnan, Pathway Technologies
   S. Agrawal, University of Delaware
   R. Venugopal, Pathway Technologies, Inc.
   M. Demeri, Ford Research Lab
3. ***Evaluating the Performance of Assistive Robotic Systems***
   H. Yanco, University of Massachusetts
4. ***Measuring Classifier Intelligence***
   J. DeLeo, National Institutes of Health
5. ***Problems of Performance Measurement in Locally-Organized Systems***
   V. Stefanuk, Russian Academy of Science

Session 1M2 – Performance of Multiple Agents - Lecture Room B

Chairs: E. Grant, M. Fields

1. ***A Control Scheme for Measure of Performance and Efficiency of Tactical Cooperative Robots***
   A. Shirkohodaei, Tennessee State University
2. ***Competitive Relative Performance Evaluation of Neural Controllers for Competitive Game Playing with Teams of Real Mobile Robots***
   A. Nelson, North Carolina State University
   E. Grant, North Carolina State University
   T. Henderson, University of Utah
3. ***Representing Ground Robotic Systems in Battlefield Simulations***
   M. Fields, Army Research Laboratory
4. ***NICCI: A Multiagent Cognitive Formation***
   E. Davidowicz, US Army, CECOM

12:00 – 1:00 – Lunch (and Meeting of Advisory Board)

1:00 – 2:00 – Plenary Lecture I-1 – Green Auditorium

James Albus, National Institute of Standards & Technology
Metrics and Performance Measures for Intelligent Unmanned Ground Vehicles

2:00 – 2:15 – Coffee Break

2:15 – 4:45 – Afternoon Sessions 1P

Session 1P1 – Performance of Mobility Systems – Lecture Room A

Chairs: E. Messina, Y. Zhang

1. ***A Simulation Framework for Evaluating Mobile Robots***
   S. Balakirsky, National Institute of Standards & Technology
   E. Messina, National Institute of Standards & Technology
2. ***Evaluating the Performance of a Vehicle Pose Measurement System***
   H. Scott, National Institute of Standards & Technology
   S. Szabo, National Institute of Standards & Technology
3. ***Performance Evaluation of Road Detection and Tracking Algorithms***
   D. Dufourd, DGA, Centre Technique d'Arcueil
   A. Digalarrondo, DGA, Centre Technique d'Arcueil
4. ***A Platform for Studying Locomotion Systems: Modular Reconfigurable Robots***
   Y. Zhang, Palo Alto Research Center
   C. Eldershaw, Palo Alto Research Center
   M. Yim, Palo Alto Research Center
   K. Roufas, Palo Alto Research Center
   D. Duff, Palo Alto Research Center

Session 1P2 – Performance of Planning Systems – Lecture Room B

Chairs: A. Meystel, W. Van Wezel

1. ***Performance of Planning Systems***
   A. Meystel, Drexel University and *National Institute of Standards & Technology*
2. ***Lower Bounds for Evaluating Schedule Performance in Flexible Job Shops***
   I. Kacem, Laboratoire d'Automatique et Informatique de Lille
   S. Hammadi, Laboratoire d'Automatique et Informatique de Lille
   P. Borne, Laboratoire d'Automatique et Informatique de Lille
3. ***Performance Characteristics of Planning Actors***
   W. Van Wezel, University of Groningen
   R. Jorna, University of Groningen
4. ***Global Optimization via SPSA***
   J. Maryak, Johns Hopkins University
   D. Chin, Johns Hopkins University

4:45 – 5:00 – Coffee Break

5:00 – 6:00 – General Discussion Panel 1 – Green Auditorium

What is the role of Ontology in Performance Evaluation?

Chairs: L. Pouchard, Oak Ridge National Laboratory
C. Schlenoff, National Institute of Standards & Technology

Participants: L. Pouchard, Oak Ridge National Laboratory
C. Schlenoff, National Institute of Standards & Technology
E. Dawidowicz, US Army, CECOM
M. Gruninger, National Institute of Standards and Technology
L. Welsch, National Institute of Standards and Technology
A. Meystel, Drexel University and National Institute of Standards & Technology

7:00 – 10:30 – **Reception** (At the Hotel)

Special presentation *Humanoids as a Testbed for Measuring Performance* with videoclips from RoboCup2002, Professor Minoru Asada, Osaka University, Japan

Wednesday, August 14, 2002

8:00 – 9:00 – Plenary Lecture II-1 – Green Auditorium

David B. Fogel, Natural Selection, Inc.
Evolving Solutions that are Competitive with Humans

9:00 – 9:30 – Coffee Break

9:30 – 12:00 – Morning Sessions 2M

Session 2M1 – Uncertainty of Representation I – Lecture Room A

Chairs: C. Landauer, J. Gunderson

1. ***Evaluation Methods for Human-System Performance of Intelligent Systems***
   J. Scholtz, National Institute of Standards and Technology
2. ***Lifelike Robotic Collaboration Requires Lifelike Information Integration***
   R. Cottam, The Evolutionary Processing Group
   W. Ranson, The Evolutionary Processing Group
   R. Vounckx, The Evolutionary Processing Group
3. ***Integrating Effective Planning Horizons into an Intelligent Systems Architecture***
   J. Gunderson, Gunderson & Gunderson, Inc.
   L. Gunderson, Gunderson & Gunderson, Inc.
4. ***Mobile Robot Pose Tracking for Performance Analysis***
   A. Lytle, National Institute of Standards and Technology
   K. Saidi, National Institute of Standards and Technology
   W. Stone, National Institute of Standards and Technology
   M. Shneier, National Institute of Standards and Technology
5. ***An Uncertainty Propagation Architecture for the Localization Problem***
   A. Clerentin, Centre de Robotique, d'Electrotechnique et d'Automatique
   L. Delahoche, Centre de Robotique, d'Electrotechnique et d'Automatique
   E. Brassart, Centre de Robotique, d'Electrotechnique et d'Automatique
   C. Cauchois, Centre de Robotique, d'Electrotechnique et d'Automatique

Session 2M2 – Performance of Robots in Hazardous Domains – Lecture Room B

Chairs: A. Jacoff, C. Norman

1. ***Lessons Learned: Experiences from the Rescue Robot Competition at RoboCup 2002***
   A. Jacoff, National Institute of Standards & Technology
   B. Weiss, National Institute of Standards & Technology
2. ***Derived Performance Metrics and Measurements Compared to Field Experience for the Packbot***
   T. Frost, iRobot
   C. Norman, iRobot
   S. Pratt, iRobot
   B. Yamauchi, iRobot
   B. McBride, South West Research Institute
   G. Peri, South West Research Institute
3. ***Intelligent Robots for Use in Hazardous DOE Environments***
   D. Bruemmer, INEEL
   J. Marble, INEEL
   D. Dudenhoeffer, INEEL
   M. Anderson, INEEL
   M. McKay, INEEL

12:00 – 1:00 – Lunch

1:00 – 2:00 Plenary II-2 – Green Auditorium

Walter Freeman, University of California, Berkeley
On Communicating with Semantic Machines

2:00 – 2:15 – Coffee Break

2:15 – 4:45 – Afternoon Sessions 2P

Session 2P1 – Modeling Intelligence – Lecture Room A

Chairs: L. Arata, S. Wallace

1. ***Modeling Interactive Intelligences***
   L. Arata, Quinnipiac University
2. ***Structured Approach to the Intelligent System Design***
   L. Polyakov, Globe Institute of Technology
3. ***Semiotic Fundamentals of Information Processing in Human Brain***
   L. Perlovsky, Air Force Research Laboratory
4. ***Intelligence and Behavioral Boundaries***
   S. Wallace, University of Michigan
   J. Laird, University of Michigan

Session 2P2 – Modeling of Mind – Lecture Room B

Chairs: S. Adams, R. Cottam

1. ***Beyond the Turing Test: Performance Metrics for Evaluating a Computer Simulation of the Human Mind***
   N. Alvarado, IBM Corporation
   S. Adams, IBM Corporation
   S. Burbeck, IBM Corporation
   C. Latta, IBM Corporation
2. ***Experimental Evaluation of Subject Matter Expert-oriented Knowledge Base Authoring Tools***
   R. Schrag, SRI International
   M. Pool, IET, Inc
   V. Chaudhri, SRI International
   R. Kahlert, Cycorp, Inc.
   J. Powers, IET, Inc
   P. Cohen, University of Massachussetts
   J. Fitzgerald, IET, Inc
   S. Mishra, SRI International
3. ***A Task Domain for Combining and Evaluating Robotics and Cognitive Modeling Techniques***
   N. Cassimatis, Naval Research Laboratory
   G. Trafton, Naval Research Laboratory
   A. Schultz, Naval Research Laboratory
   M. Bugajska, Naval Research Laboratory
   W. Adams, Naval Research Laboratory
4. ***Modelling, Self and Consciousness: Further Perspectives of AI Research***
   R. Sanz, Universidad Politecnicia de Madrid, Spain
   A. Meystel, Drexel University

4:45 – 5:00 – Coffee Break

5:00 – 6:00 – General Discussion Panel 2 – Green Auditorium

Information Access Technology: Does it Concern Mechanisms of Mind and Intelligence?

Chairs: M. Herman, National Institute of Standards & Technology
Participants: A. Meystel, Drexel University, National Institute of Standards & Technology
J. Cugini, National Institute of Standards & Technology
J. Garofolo, National Institute of Standards & Technology
J. Phillips, National Institute of Standards & Technology and DARPA

6:00 – 7:30 – Break

7:30 – 10:30 Banquet (At the Hotel)

Banquet Speech

Lofti Zadeh, University of California, Berkeley

In Quest of Performance Metrics for Intelligent Systems – A Challenge that Cannot be Met with Existing Methods

Thursday, August 15, 2002

8:00 – 9:00 – Plenary Lecture III-1 – Green Auditorium

John Blitch, Center for Robot-Assisted Search and Rescue

Robot Intelligence for Tunneling and Confined Space Search and Rescue

9:00 – 9:30 – Coffee Break

9:30 – 12:00 – Morning Sessions 3M

Session 3M1 – Measuring Intelligence – Lecture Room A

Chairs: J. Horst, C. Schlenoff

1. *Metrics, Schmetrics! How the Heck do you Determine a UAV's Autonomy Anyway?*
   B. Clough, Air Force Research Laboratory, Wright-Patterson AFB
2. *Towards Measuring the Performance of Architectural Components of Autonomous Vehicular Systems*
   C. Schlenoff, National Institute of Standards & Technology
   L. Welsch, National Institute of Standards & Technology
   R. Madhavan, National Institute of Standards & Technology
   N. Zimmerman, National Institute of Standards & Technology
3. *A Native Intelligence Metric for Artificial Systems*
   J. Horst, National Institute of Standards & Technology
4. *Dimensions of Intelligent Systems*
   G. Berg-Cross, SLAG, Inc.

Session 3M2 – Grouping: A Core Procedure of Intelligence – Lecture Room B

Chairs: S. Ramaswamy, R. Bialczak

1. *A New Classification of Information: A Step on the Road to Interpretability*
   L. Reeker, National Institute of Standards & Technology
   A. Jones, National Institute of Standards & Technology
2. *Refactored Characteristics in Intelligent Computing Systems*
   C. Landauer, The Aerospace Corporation
   K. Bellman, The Aerospace Corporation
3. *Software Design and Testing using Petri Nets: A Case Study Using a Distributed Simulation Software System*
   S. Ramaswamy, Tennessee Technological University
   R. Neelakantan, Tennessee Technological University
4. *Comparison Methodology for Robotic Operator Control Units*
   R. Bialczak, TSI, Inc.
   J. Nida, TSI, Inc.
   B. Pettitt, TSI, Inc.
   M. Kalphat, STRICOM

12:00 – 1:00 – Lunch

1:00 – 2:00 – Plenary Lecture III-2 – Green Auditorium

Bernie Zeigler, University of Arizona
Scalability Considerations in Measuring Intelligence: Insights from Modeling and Simulation

---

2:00 – 2:15 – Coffee Break

2:15 – 4:45 Afternoon Sessions 3P

Session 3P1 – Uncertainty in Representation II – Lecture Room A

Chairs: D. Chin, R. Maarfi

1. ***Multiple Neural Network Model Interpolation***
   D. Chin, Johns Hopkins University, Applied Physics Laboratory
   A. Biondo, Johns Hopkins University, Applied Physics Laboratory
2. ***A Three-Tier Communication and Control Structure for the Distributed Simulation of an Automated Highway System***
   R. Maarfi, Tennessee Technological University
   E.L. Brown, Tennessee Technological University
   S. Ramaswamy, Tennessee Technological University
3. ***Tessellating and Searching in Uncertain State Spaces***
   A. Meystel, Drexel University
   A. Bathija, Drexel University
4. ***Autonomy and Socialization***
   K. Bellman, The Aerospace Corporation

Session 3P2 – Towards Universal Planning/Control Systems – Lecture Room B

Chairs: A. Meystel, P. Davis

1. ***A Sketch of Multiresolutional Decision Support Systems Theory***
   A. Meystel, Drexel University
2. ***Motivated Metamodels***
   P. K. Davis, RAND
   J. H. Bigelow, RAND
3. ***On the Role of Quality Attributes in Specifying Software/System Architecture for Intelligent Systems***
   H. Sarjoughian, Arizona State University
4. ***Uncertain Predictions of Flow and Transport in Random Porous Media: The Implications for Process Planning and Control***
   S. Orr, MRDS, Inc.

4:45 – 5:00 – Coffee Break

5:00 – 6:00 – General Discussion Panel 3 – Green Auditorium
Government Support of Research in Performance Evaluations

Chair: E. Messina, National Institute of Standards & Technology

Participants: D. Gage, DARPA
J. Albus, National Institute of Standards & Technology
C. Shoemaker, US Army, ARL
E. Dawidowicz, US Army, CECOM
B. Clough, WPAFB
F. Darema, National Science Foundation
J. Overholt, TACOM
D. Dudenhoeffer, INEEL, DOE

APPENDIX B
FINAL PARTICIPANTS LIST

Final Participants' List

Performance Metrics for Intelligent Systems

August 13-15, 2001

National Institute of Standards and Technology (NIST), Gaithersburg, MD

Sam Adams
IBM Research
105 Chilcott Lane
Apex, NC 27502 USA
Telephone: 919/254-0736
Fax: 919/362-3929
Email: ssadams@us.ibm.com

James Albus
NIST
100 Bureau Dr., Mail Stop 8230
Gaithersburg, MD 20899-8230 USA
Telephone: 301/975-3418
Fax: 301/990-9688
Email: james.albus@nist.gov

Nancy Alvarado
IBM Research
8 Iron Bark Way
Irvine, CA 92612 USA
Telephone: 949/387-6010
Email: alvadado@psy.ucsd.edu

Luis Arata
Quinnipiac Univ.
Hamden, CT 06515 USA
Telephone: 203/582-8658

M. Clerentin Arnaud
Credit Agricole SA
Amiens, FRANCE

Minoru Asada
Osaka Univ.
2-1 Yamadaoka
Suita OSA 565-0871, JAPAN
Telephone: 81/6 6879-7347
Fax: 81/6 6879 7348
Email: asada@ams.eng.osaka-u.ac.jp

Stephen Balakirsky
NIST
100 Bureau Dr., Mail Stop 8230
Gaithersburg, MD 20899-8230 USA
Telephone: 301/975-4791
Fax: 301/990-9688
Email: stephen.balakirsky@nist.gov

Kirstie Bellman
Aerospace Integration Science Ctr.
P.O. BOx 92957
Los Angeles, CA 90009 USA
Telephone: 310/336-2191
Email: bellman@aero.org

Gary Berg-Cross
Knowledge Strategies
13 Atwell Court
Potomac, MD 20854 USA
Email: gberg-cross@yahoo.com

Robert Bialczak
Science & Engineering Svcs.
1584 W. Lincoln Trail
Radcliff, KY 40160 USA
Telephone: 270/351-2035

John Blitch
Ctr. for Robot-Assisted Search
P.O. 270346
Littleton, CO 80127 USA
Telephone: 703/431-0082
Fax: 703/697-7607
Email: jblitch@worldnet.att.net

Zvi Boger
Optimal-Ind Neural System
261 Congressional La.
Ste. 319
Rockville, MD 20852 USA
Telephone: 301/984-8648
Fax: 301/984-8648

Nick Cassimatis
Naval Research Lab
4555 Overlook Ave.
Code 5513
Washington, DC 20375 USA
Telephone: 617/504-9497
Email: nc@alum.mit.edu

Daniel Chin
Johns Hopkins Univ.
11100 Johns Hopkins
Laurel, MD 20723 USA
Telephone: 443/778-4956
Fax: 443/778-6661
Email: Daniel.Chin@jhaupl.edu

Bruce Clough
US Air Force Research Laboratory
AFRL/VACC
2130 8th St.
WPAFB, OH 45433-7542 USA
Telephone: 937/255-2831
Fax: 937/255-8297
Email: bruce.clough@wpafb.af.mil

Betsy Constantine
Context Systems
124 Indian Hill Rd.
Carlisle, MA 01741 USA
Telephone: 978/369-0057
Email: bconstantine@earthlink.net

Ron Cottam
IMEC
EVOL-ETRO-VUB
Pleincaan 2 B 1050, BELGIUM
Telephone: 32/2 629 2933
Email: ron@cottam.be

John Cugini
NIST
100 Bureau Dr., Mail Stop 8940
Gaithersburg, MD 20899-8940 USA
Telephone: 301/975-3248
Fax: 301/975-5287
Email: cus@nist.gov

Edward Dawidowicz
U.S. Army, CECOM, RDEC
Bldg. 2700
Mayer Center
Fort Monmouth, NJ 07702 USA
Telephone: 732/427-4122
Fax: 732/427-3440
Email:
edward.dawidowcz@mail1.monmouth.army.mil

James DeLeo
NIH/Clinical Ctr.
10 Center Dr.
Bldg. 10, Rm. 1C290
Bethesda, MD 20892 USA
Telephone: 301/496-3848
Fax: 301/496-3009
Email: deleoj@mail.nih.gov

Don Dudenhoeffer
INEEL
P.O. Box 1625
MS 3605
Idaho Falls, ID 83415-3605 USA
Telephone: 208/526-0700
Fax: 208/526-2777

Delphine Dufourd
DGA/Centre Technique d'Ar
16bis avenue Prieur de la Cote D'Or
DT/GIP
Arcueil 94114, FRANCE
Telephone: 33/1 42 31 9707
Fax: 33/1 42 31 9964
Email: Delphine.Dufourd@etca.fr

Sanjiv Dungrani
U.S. Army Tank
6501 E. 11 Mile Rd.
M/S 264
Warren, MI 48397 USA
Telephone: 586/574-5696
Fax: 586-574-5008
Email: dungrans@tacom.army.mil

Maryanne Fields
U.S. Army Research Laboratory
ATTN: AMSRL-WM-BF
Aberdeem Proving Grd., MD 21005-5066
USA
Telephone: 410/278-6675
Fax: 410/278-9078
Email: mafld@arl.army.mil

Robert Finkelstein
Robotic Technology, Inc.
11424 Palatine Drive
Potomac, MD 20854 USA
Telephone: 301/983-4194

David Fogel
Natural Selection, Inc.
3333 N. Torrey Pine Court, Ste. 200
La Jolla, CA 92037 USA
Telephone: 858/455-6449
Fax: 858/455-1560
Email: dfogel@natural-selection.com

Walter Freeman
Univ. of California, Berkeley
MCB-N LISA 142
Berkely, CA 94704-1200 USA
Telephone: 510/642-4220
Fax: 510/643-6971
Email: wfreeman@socrates.berkeley.edu

Douglas Gage
DARPA
3701 N. Fairfax Dr.
Arlington, VA 22203 USA
Telephone: 703/696-1122

Reuven Granot
Israel MoD, MAFAT
Hakirya Tel Aviv
Tel Aviv, ISRAEL
Telephone: 972/534 10193
Fax: 972/482 21128
Email: rgranot@tx.technion.ac.il

Eddie Grant
NC State Univ.
411 Daniels Hall
CRIM, ECE Dept.
Raleigh, NC 27695 USA
Telephone: 919/515-7354
Fax: 919/515-5523
Email: egrant@eos.ncsu.edu

James Gunderson
Gunderson & Gunderson, Inc.
1407 Forest Ridge Rd.
Charlottesville, VA 22903 USA
Telephone: 434/982-8091

Louise Gunderson
Gunderson & Gunderson, Inc.
1407 Forest Ridge Rd.
Charlottesville, VA 22903 USA
Telephone: 434/982-8091
Email: lfg4a@virginia.edu

Tom Haug
U.S. Army Research Laboratory
ATTN: AMSRL-WM-BF
Aberdeen Proving Grd., MD 21005-5066
USA
Telephone: 410/278-6178
Fax: 410/278-9078
Email: haug@arl.army.mil

Thomas Henderson
Univ of Utah
50 S. Central Campus, Rm. 3190
Salt Lake City, UT 84112 USA
Telephone: 801/581-3601
Fax: 801/581-5843
Email: tch@cs.utah.edu

John Horst
NIST
100 Bureau Dr., Mail Stop 8230
Gaithersburg, MD 20899-8230 USA
Telephone: 301/975-3430
Fax: 301/990-9688
Email: john.horst@nist.gov

Adam Jacoff
NIST
100 Bureau Dr., Mail Stop 8230
Gaithersburg, MD 20899-8230 USA
Telephone: 301/975-4235
Email: adam.jacoff@nist.gov

Hank Johannson
PICO Technologies
9112 Balmoral Mews Square
Windermere, FL 34786-6208 USA
Telephone: 407/290-5991
Email: hjohannson@aol.com

Imed Kacem
Lail Ecole Centrale De Lille
Cite Scientifique
Vinve d'Ascq 59651, FRANCE
Telephone: 33/203 35452
Fax: 333/203 35499
Email: imed.kacem@ec-lille.fr

Lee Karlowitz
Dept. of Defense
9800 Savage Rd.
Ft. Meade, MD 20755 USA
Telephone: 443/479-8821

Christopher Landauer
Aerospace Integration Science Ctr.
P.O. Box 92957
Los Angeles, CA 90009 USA
Telephone: 310/336-1361
Email: cal@aero.org

Karlowitz Lee
Dept. of Defense
9800 Savage Rd.
Ft. Meade, MD 20755 USA
Telephone: 443/479-8821
Fax: 443/479-8888
Email: lpkarlo@fggm.osis.gov

A. Scott Lewis
PA State Univ.
Applied Research Laboratory
State College, PA 16802 USA
Telephone: 814/865-0962
Fax: 814/863-7841
Email: as1103@psu.edu

Ron Lumia
Univ of New Mexico
Dept of Mechanical E
Albuquerque, NM 87122 USA
Telephone: 505/272-7155

Raj Madhavan
NIST
100 Bureau Dr., Mail Stop 8230
Gaithersburg, MD 20899-8230 USA
Telephone: 301/975-2865
Fax: 301/990-9688
Email: raj.madhavan@nist.gov

Elena Messina
NIST
100 Bureau Dr., Mail Stop 8230
Gaithersburg, MD 20899-8230 USA
Telephone: 301/975-3510
Fax: 301/990-9688
Email: elena.messina@nist.gov

Alex Meystel
Drexel Univ.
P.O. Box 2404
Bala Cynwyd, PA 19004 USA
Telephone: 610/667-2562
Fax: 610/667-2008
Email: alex@meystel.com

Andrew Nelson
NC State Univ.
411 Daniels Hall
CRIM, Dept. ECE
Raleigh, NC 27695 USA
Telephone: 919/515-7452
Fax: 919/515-5523
Email: alnelso2@eos.ncsu.edu

Jonathan Nida
Science & Eng Svcs., Inc.
1584 W. Lincoln Trail
Radcliff, KY 40160 USA
Telephone: 270/351-2035

Christopher Norman
IRobot Corp.
22 McGrath Hgwy.
Somerville, MA 02145 USA
Telephone: 617/629-0055

Shlomo Orr
MRDS, Inc.
5900 W 25th Ave.
Kennewick, WA 99338 USA
Telephone: 509/736-3111

Jim Overholt
U.S. Army TACOM
AMSTA-TR-R/263
Warren, MI 48397 USA
Telephone: 586/574-8618

Brian Pettitt
Tech. Service Associates
3361 Rouse Rd., Ste. 240
Orlando, FL 32828 USA
Telephone: 270/258-1262

Michael Pool
IET, Inc.
1911 North Myer Dr., Ste. 600
Arlington, VA 22209 USA
Telephone: 703/841-3500

Line Pouchard
Oak Ridge Nat'l. Lab
1 Bethel Valley Rd., MS 6367
Oak Ridge, TN 37983 USA
Telephone: 865/574-6125
Fax: 865/574-0680
Email: pourchardlc@ornl.gov

Srinivasan Ramaswamy
Tennessee Tech. Univ.
Box 5101
Computer Science Dept.
Cookeville, TN 38501 USA
Telephone: 931/372-3691
Fax: 931/372-3686
Email: srini@acm.org

Muralidhar Rangaswamy
Airforce Research Laboratory
80 Scott Dr.
Hanscom Air Force Base, MA 01731-2909
USA
Telephone: 781/377-3446
Fax: 781/377-8984
Email:
muralidhar.rangaswamy@hanscom.af.

John Reid
Deere & Co.
1 John Deere Place
Moline, IL 61265 USA
Telephone: 309/765-3786
Fax: 309/765-3807
Email: ReidJohnF@JohnDeere.com

William Romans
Deere & Co.
John Deere Tech. Ctr.
Moline, IL 61265 USA
Telephone: 309/765-3743
Fax: 309/765-3807
Email: romans.william.r@johndeere.com

Mark Rosenblum
PercepTek
9892 Titan Park Circ.
Littleton, CO 80125 USA
Telephone: 720/344-1037

Ricardo Sanz
Universidad Politecnica de Madrid
Jose Gutierrez Abasc2
Madrid 28006, SPAIN
Telephone: 34/91 3363061
Fax: 34/91 3363010
Email: Ricardo.Sanz@etsii.upm.es

Hessam Sarjoughian
Arizona State Univ.
Computer Scie. & Engineering
Tempe, AZ 85287 USA
Telephone: 480/965-3983

Craig Schlenoff
NIST
100 Bureau Dr., Mail Stop 8230
Gaithersburg, MD 20899-8230 USA
Telephone: 301/975-3456
Fax: 301/990-9688
Email: craig.schlonof@nist.gov

Jean Scholtz
NIST
100 Bureau Dr., Mail Stop 8940
Gaithersburg, MD 20899-8940 USA
Telephone: 301/975-2520
Fax: 301/975-5287
Email: jean.scholtz@nist.gov

Alan Schultz
NRL/ONR
4555 Overlook Ave.
Washington, DC 20375 USA
Telephone: 202/767-2684

Richard Sherry
Penn State Univ.
School of Information Scie. & Tech.
001 Thomas Building
University Park, PA 16802 USA
Telephone: 814/865-4455
Email: rrsherry@psu.edu

James Stover
PA State Univ.
Applied Research Laboratory
State College, PA 16802 USA
Telephone: 814/863-4104
Fax: 814/863-7841
Email: jj5@psu.edu

Anathalcris Suri
Pathway Technologies, Inc.
510 Township Line Rd., Ste. 110
Blue Bell, PA 19422 USA
Telephone: 215/542-4925

Wendell Sykes
Context Systems
124 Indian Hill Rd.
Carlisle, MA 01741 USA
Email: sykes@alum.mit.edu

Walt Truszkowski
NASA/Goddard Space Flight Ctr.
Code 588
Greenbelt, MD 20771 USA
Telephone: 301/286-8821
Email: Walt.Truszkowski@gsfc.nasa.gov

Lefteri Tsoukalas
Purdue Univ.
1290 Nuclear Engineering Bldg.
West Lafayette, IN 47907-1290 USA
Telephone: 765/494-5742
Fax: 765/494-9570
Email: nehead@ecn.purdue.edu

Scott Wallace
Univ. of Michigan
1101 Beal Ave.
Ann Arbor, MI 48109 USA
Telephone: 734/764-4343
Fax: 734/763-1260
Email: swallace@umich.edu

Harold Warner
NAVAIR
48110 Shaw Rd.
Bldg. 2187, Ste. 228
Patuxent River, MD 20670 USA
Telephone: 301/342-9284
Fax: 301/342-9305
Email: warnerhd@navair.navy.mil

Brian Weiss
NIST
100 Bureau Dr., Mail Stop 8230
Gaithersburg, MD 20899-8230 USA
Telephone: 301/975-4373
Email: brian.weiss@nist.gov

Roger Wells
G Wells & Associates
8611 Etta Dr.
Springfield, VA 22152 USA
Telephone: 703/451-8409
Email: c.roger.wells@cox.net

Lawrence Welsch
NIST
100 Bureau Dr., Mail Stop 8220
Gaithersburg, MD 20899-8220 USA
Telephone: 301/975-3198
Email: lawrence.welsch@nist.gov

W. Van Wezel
Univ. of Groningen
Landleven 5
Groningen 9700 AV, NETHERLAND
Telephone: 31/50 3637181
Email: w.m.c.van.wezel@bdk.rug.nl

Holly Yanco
UMass Lowell
One University Ave., CS Dept.
Lowell, MA 01854 USA
Telephone: 978/934-3642
Fax: 978/934-3551
Email: holly@cs.uml.edu

Lofti Zadeh
Univ. of California, Berkeley
Berkely, CA 94720-1776 USA
Telephone: 510/642-4959
Fax: 510/652-1712
Email: zadeh@cs.berkeley.edu

Bernie Zeigler
Univ. of Arizona
ECE Dept.
Tucson, AZ 85721 USA
Telephone: 520/621-2108
Fax: 520/621-3862
Email: zeigler@ece.orizona.edu

Ying Zhang
Palo Alto Research Ctr.
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA
Telephone: 650/812-4654
Fax: 650/812-4334
Email: yzhang@parc.com

APPENDIX C
AUTHOR INDEX

NIST *Technical Publications*

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.
Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.
Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.
National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bimonthly for NIST by the American Institute of Physics (AIP). Subscription orders and renewals are available from AIP, P.O. Box 503284, St. Louis, MO 63150-3284.
Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.
Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.
Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program in support of the efforts of private-sector standardizing organizations.

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*
Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).
NIST Interagency or Internal Reports (NISTIR)—The series includes interim or final reports on work performed by NIST for outside sponsors (both government and nongovernment). In general, initial distribution is handled by the sponsor; public distribution is handled by sales through the National Technical Information Service, Springfield, VA 22161, in hard copy, electronic media, or microfiche form. NISTIR's may also report results of NIST projects of transitory or limited interest, including those that will be published subsequently in more comprehensive form.

U.S. Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001