

Technical Report 395

LEVEL II

1
31

ADA 084730

PREVENTION AND REMEDIATION OF HUMAN INPUT ERRORS IN ADP OPERATIONS

Douglas J. Mace, Paul C. Harrison, Jr.,
and Edmond L. Seguin
Institute for Research

HUMAN FACTORS TECHNICAL AREA

DTIC
ELECTED
MAR 12 1980
S A D



U. S. Army

Research Institute for the Behavioral and Social Sciences

August 1979

Approved for public release; distribution unlimited.

80 3 12 011

DDC FILE COPY

**U. S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES**

**A Field Operating Agency under the Jurisdiction of the
Deputy Chief of Staff for Personnel**

JOSEPH ZEIDNER
Technical Director

WILLIAM L. HAUSER
Colonel, U S Army
Commander

Research accomplished under contract
for the Department of the Army

Institute for Research

NOTICES

DISTRIBUTION: Primary distribution of this report has been made by ARI. Please address correspondence
concerning distribution of reports to: U. S. Army Research Institute for the Behavioral and Social Sciences.
ATTN: PERI-P, 5001 Eisenhower Avenue, Alexandria, Virginia 22333.

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to
the U. S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this report are not to be construed as an official Department of the Army position,
unless so designated by other authorized documents.

18 ARI

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report 395	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PREVENTION AND REMEDIATION OF HUMAN INPUT ERRORS IN ADP OPERATIONS		5. APPROVAL NUMBER Final Report
6. AUTHOR(s) D. J. Mace, P. C. Harrison, Jr. L. Seguin		7. PERFORMING ORGANIZATION NUMBER ---
8. CONTRACT OR GRANT NUMBER(s) DAHC19-78-C-0017		9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 20763743A774
10. CONTROLLING OFFICE NAME AND ADDRESS Combined Arms Combat Developments Activity Fort Leavenworth, KS 66027		11. REPORT DATE August 1979
12. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) U.S. Army Research Institute for the Behavioral and Social Sciences 5001 Eisenhower Avenue, Alexandria, VA 22333		13. NUMBER OF PAGES 110
14. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		15. SECURITY CLASS. (of this report) Unclassified
16. DECLASSIFICATION/DOWNGRADING SCHEDULE 12 122		17. DISTRIBUTION STATEMENT (of the abstract entered in Block 10, if different from Report) ---
18. SUPPLEMENTARY NOTES Monitored technically by Raymond G. Sidorsky, Human Factors Technical Area, ARI.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Input errors Computer input errors Human computer interaction Man-computer interaction		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report classifies types of human errors made while entering data into automatic data processing (ADP) systems. Input error causes, effects, and prevention are discussed specifically in terms of tactical ADP systems for automated battlefield information systems. The multi-attribute utility measurement approach described here provides a method for evaluating costs and benefits of different error-reduction techniques.		

DD FORM 1 JAN 73 EDITION OF 1 NOV 68 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

180200

PREVENTION AND REMEDIATION OF HUMAN INPUT ERRORS IN ADP OPERATIONS

Douglas J. Mace, Paul C. Harrison, Jr.,
and Edmond L. Seguin
Institute for Research

Raymond C. Sidorsky, Team Chief

Submitted by:
Edgar M. Johnson, Chief
HUMAN FACTORS TECHNICAL AREA

Accession For	
HIS GRAM	
DDC TAB	
Unannounced	
Justification	
By	
Distribution	
Availability Codes	
Dist	Avail and/or special

Approved by:

Frank J. Harris, Acting Director
ORGANIZATIONS AND SYSTEMS
RESEARCH LABORATORY

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES
5001 Eisenhower Avenue, Alexandria, Virginia 22333

Office, Deputy Chief of Staff for Personnel
Department of the Army

August 1979

Army Project Number
20763743A774

Information Processing
and Display

Approved for public release; distribution unlimited.

ARI Research Reports and Technical Reports are intended for sponsors of R&D tasks and for other research and military agencies. Any findings ready for implementation at the time of publication are presented in the last part of the Brief. Upon completion of a major phase of the task, formal recommendations for official action normally are conveyed to appropriate military agencies by briefing or Disposition Form.

FOREWORD

The Human Factors Technical Area is concerned with aiding users/operators to cope with the ever increasing complexity of the man-machine systems being designed to acquire, transmit, process, disseminate, and utilize tactical information on the battlefield. The research is focused on interface problems and interactions within command and control centers and is concerned with such areas as topographic products and procedures, tactical symbology, user-oriented systems, information management, staff operations and procedures, and sensor systems integration and utilization.

The U.S. Army is turning increasingly toward the use of automated battlefield systems to meet anticipated mission requirements. More than 70 separate automated systems are currently in a developmental or concept definition phase. However, the inability of operators and users to interact effectively with many of the current automated data systems has severely reduced their effectiveness. High error rates, low input rates, and inappropriately structured outputs have largely offset the potential benefits of automation. Many of these errors occur at the human-computer interface during data entry. The present publication provides a classification schema for categorizing human errors produced at the operator/computer interface in battlefield automated systems. TOS (Tactical Operations System) was used as a focus to identify the causes of each type of error, to suggest error remediation techniques, and to provide procedures for assessing the cost benefits of alternative error reduction techniques.

Research leading to improved design and procedures at the human-computer interface is conducted as an in-house effort augmented through contracts with organizations selected for their unique capabilities and facilities for behavioral research and analysis of automated information processing systems and operations. The present study was conducted by personnel of the Institute for Research under Contract DAHC19-78-C-0017. The effort is responsive to requirements of Army Project 2Q763743A774 and special requirements of the Combined Arms Combat Developments Activity, Fort Leavenworth, Kans. Special requirements are contained in Human Resources Need 79-104, Interactive Procedures for Data Inputting, Organization, Retrieval and Purge, and 79-111, Alternative Input Techniques for Tactical Data Systems.


JOSEPH ZEIDNER
Technical Director

PREVENTION AND REMEDIATION OF HUMAN INPUT ERRORS IN ADP OPERATIONS

BRIEF

Requirement:

To reduce the number and seriousness of the data entry errors made by operators and users of automated battlefield information processing systems.

Procedure:

The operator requirements and procedures of a representative sample of automated data processing systems were examined and a classification developed for categorizing human errors at the operator/computer interface of battlefield automated systems. The analysis considered such factors as the types of errors (character level, message level, etc.), properties of errors (frequency, criticality, etc.), and the impact of the errors on system output.

Findings:

The Tactical Operations System (TOS) was used as a focus. It was determined that: (1) the basic causative factors associated with each type of input errors can be identified; (2) techniques for detecting and remedying such errors are available (although some are prohibitively expensive for most applications); and (3) a procedure exists that holds promise as a means for assessing objectively the relative cost-benefits of alternative error reduction techniques.

Utilization of Findings:

The products of this analysis will assist system proponents and designers of automated battlefield data systems to minimize the number and impact of operator input errors. An objective methodology is provided for the selection of the design features and operating procedures at the human/computer interface which best match the needs and capabilities of the anticipated users/operators with the characteristics of the system hardware/software.

TABLE OF CONTENTS

OVERVIEW	1
I. INTRODUCTION	10
A. Overview and Purpose	10
B. Operator Activities in the ADP Interface	11
C. Scope	14
II. ERROR CLASSIFICATION SCHEMA	17
A. General	17
B. Errors of Omission	18
C. Errors of Commission	24
III. SYSTEM CONTEXT FOR ERROR REMEDIATION	33
A. General	33
B. External Influences	34
C. Internal Influences	40
IV. APPROACHES TO THE DESIGN OF THE INPUT INTERFACE	44
A. Input Language	45
B. Formatting	50
C. Automated Processes	60
D. Interactive Processes	65
E. Special Data Entry Techniques	71
F. Summary	75
V. COST BENEFIT ANALYSIS	76
A. Introduction	76
B. Cost Benefit Analysis: The Context	77
C. The Cost Benefit Analyst	80
D. The Solution Space	82
E. Objectives and Suboptimization	84
F. Costs	85
G. Benefits	88
H. Multi-Attribute Utility Measurement	90
VI. APPLICATION OF THE MAUM PROCESS TO THE TOS INPUT ERROR/REMEDIATION DECISION PROBLEM: AN ILLUSTRATIVE EXAMPLE	94
REFERENCES	109

LIST OF FIGURES

Figure 1. Error Classification Schema	3
Figure 2. A Schematic Representation of the MAUM Decision Process Applied to the TOS Error Remediation Problem	9
Figure 3. Schematic Representation of Information Systems .	12
Figure 4. Error Classification Schema for Errors of Omission	19
Figure 5. Error Classification Schema for Errors of Commission	25
Figure 6. Two-Way Classification of Errors of Commission .	27
Figure 7. Typical Solution Space Showing Relative Locations of Hypothetical Alternative Remediations (A thru G)	83
Figure 8. A Schematic Representation of the MAUM Decision Process Applied to the TOS Error Remediation Problem	108

LIST OF TABLES

Table 1. Error Types, Causes, and Alternatives for Prevention and Detection	4
---	---

OVERVIEW

Objectives

The application of ADP information processing systems to tactical operations has been plagued by high errors rates. Many of these errors occur at the man-computer interface during the process of data entry. This document details the initial steps to define an approach to the analysis and solution of this problem which is methodologically practical, i.e. recognizes the constraints of the tactical environment and the complexities of the system design/development process within the military.

The major objectives of the project are as follows:

1. Develop a comprehensive classification schema for categorizing human errors at the operator/computer interface of tactical ADP systems, in general, and with particular emphasis on the major input tasks.
2. Identify the basic causative factors which contribute to the occurrence of each type of error, review and classify remediation approaches in interface design, and associate the prevention/amelioration procedures most probably effective for each type of error.
3. Develop an approach for estimating/assessing the relative contribution or importance of each type of error in degrading overall system performance.
4. Develop procedures for assessing the relative cost/benefit consequences of alternative approaches to mitigating the effects of various types of data entry level errors, and illustrate the application of the procedures to the TOS system.

Focus on Tactical ADP Systems

In the process of accomplishing these objectives it became apparent that, in order to satisfy the requirements to be practical and realistic, it was necessary to emphasize the focus upon tactical ADP systems and upon the complexity of the tradeoff process among the major system factors. To support this emphasis and to provide for its orderly consideration, a section (Chapter 3) is provided which discusses the impact of constraints derived

from consideration of the external factors of the system mission and the tactical environment upon internal system factors of hardware, software, personnel, and procedures.

Schema for Classifying Errors

A schema for classifying errors in response to technical objective 1 is discussed in Chapter 2 and reproduced here in Figure 1. The dimensions underlying the classification schema are relevant to explanations of the cause of errors and methods for their control and prevention. There are three major concepts underlying the error classification schema. First, input errors always involve either the omission of information or the input of wrong information. Second, with respect to some data elements the validity of information input can be tested automatically by the system, while the validity of information in other data elements can only be established after the fact as evidenced by the quality of information disseminated. Third, where the validity can be tested by the system some incidents of wrong information will be detected (e.g. abbreviation errors) and others will not (e.g. glossary errors). System validation can be performed on both verbal and quantitative data elements for which the causes and remediation are often quite different.

Causes of Input Errors

The causes for each type of error in response to technical objective 2 are discussed along with the error classification schema in Chapter 2. Since there were no empirical data obtained and no specific system considered, the causes identified are analytically derived and represent the range of factors which might contribute to error in any one system. The identification of the cause of error in a specific system is a complex issue which must consider all the constraints in which the system was designed to operate. One could argue, for example, that the machine causes all errors by not accepting natural language input or that operators cause all errors out of either ignorance of systems requirements or carelessness, but systems are not intended to accept everything the operator might "throw at them" anymore than users are required to speak a binary language. The causes identified in Chapter 2 and summarized in Table 1 therefore reflect a more realistic view of the constraints imposed on man-machine dialogues and suggest the most probable causes to consider when analyzing the occurrence of error in any one system.

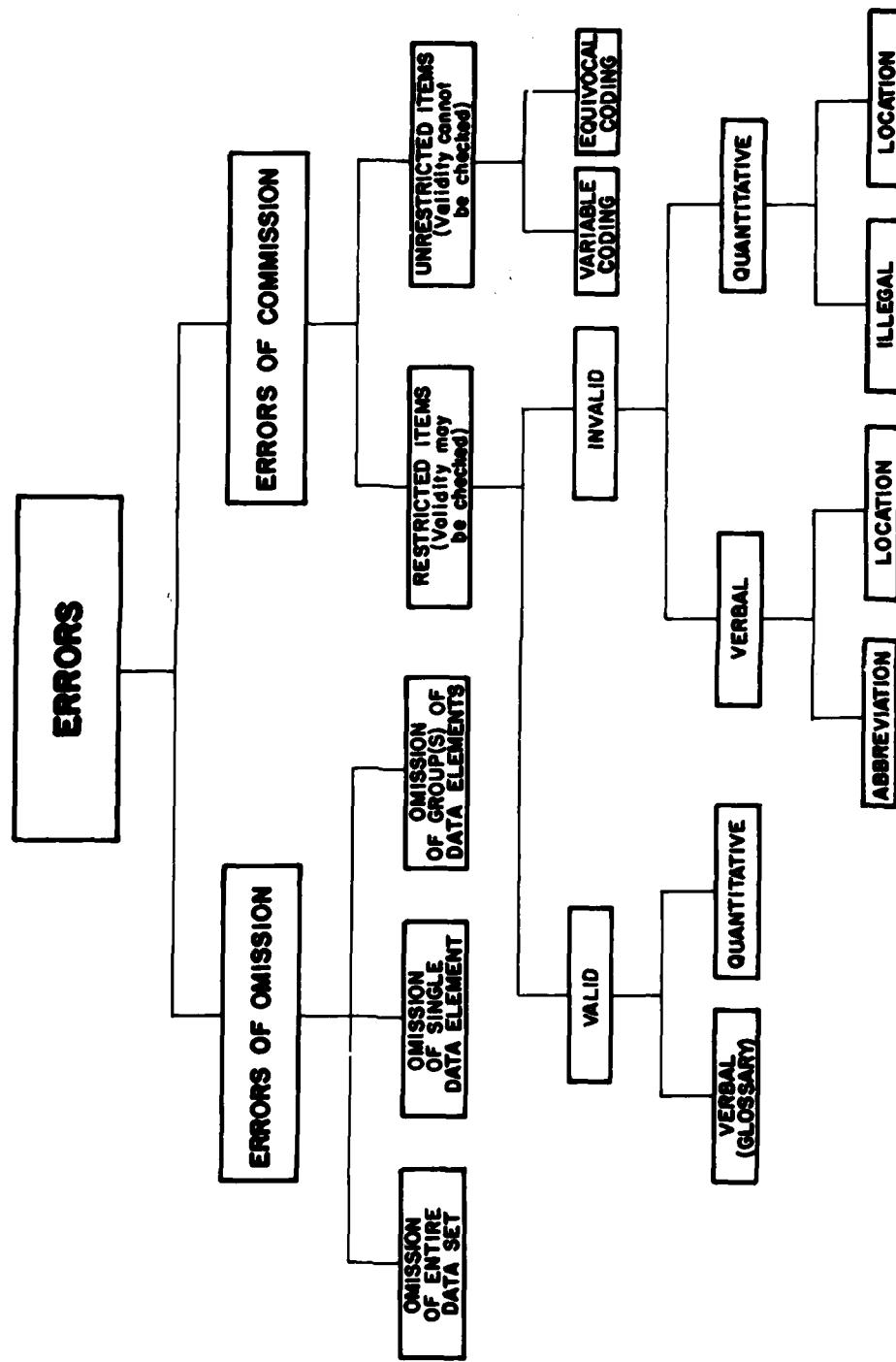


Figure 1. Error Classification Schema

Table 1. Error types, causes and alternatives for prevention and detection

ERROR TYPE	CAUSE OF ERROR	PREVENTION AND DETECTION
OMISSION OF MESSAGE SET		
	Lack of knowledge of user/operator	User/operator selection
	Incompatibility between source document and input dialogue	User/operator training
	Misplaced document	Revise procedures - horizontal distribution of input activities
	Operator failure	Revise source document formats
		Revise input formats
OMISSION OF DATA ELEMENT GROUP		
	Incompatibility between source document and input dialogue	Revise procedures - vertical distribution of input activities
	Skip a line or lose a page	Revise source document formats
		Revise input formats
OMISSION OF DATA ELEMENT		
	Improper presumption of default values	Use formats with explicit labels
	Element input into wrong location	Display default values
	Loss of place in source or pre-formatted document	Conditional error checking
	Loss of place in dialogue with system, e.g. cursor position	Conditional formatting
		Interactive dialogue
VALID CODES/RESTRICTED ITEMS		
Verbal Errors (Glossary)	Incorrect recall	Input language
	Incorrect recognition	Expanded definitions
	Transcription errors	Conditional, probabilistic, or adaptive error checking
Quantitative Errors	Incorrect scale conversions	Editing process
	Data input with incorrect scale	Formatting
	Incorrect rounding of numbers	Conditional, probabilistic, or adaptive error checking
	Careless transcription, including character transposition and decimal placement	

(cont'd)

Table 1. Error types, causes and alternatives for prevention and detection (cont'd)

ERROR TYPE	CAUSE OF ERROR	PREVENTION AND DETECTION
INVALID CODES/RESTRICTED ITEMS		
Verbal Errors - Incorrect Abbreviations	Incorrect recall Incorrect recognition Typographic mistake	Input language Menu selection Displayed codes Glossary display or "HELP" routines
Verbal Errors - Incorrect Location	Typographic mistake Format generated confusion	Formats with explicit labels Data base update Conditional formatting Question and answer dialogue Menu selection Displayed codes
Quantitative Errors - Incorrect Location	Typographic mistake Format generated confusion	Formats with explicit labels Data base update Conditional formatting Question and answer dialogue
Quantitative Errors - Illegal Entry	Incorrect scale conversion Data input with incorrect scale Incorrect rounding of numbers Carless transcription, including character transposition and decimal placement	Editing Processes Formatting Question and answer dialogue
RESTRICTED ITEMS		
Equivocal Coding (Same code used with two or more meanings)	Coding conventions inadequate Data element too short	Improper coding conventions Lengthen data element Provide glossary of previous entries Data base update Expanded definitions
Variable Coding (Two or more codes used with same meaning)	Coding conventions inadequate Mispellings Species Number of coded entries which must be recalled	Improper coding conventions Provide glossary of previous entries Data base update Expanded definitions

Approaches to Error Detection/Prevention

For each type of error, Table I also summarizes some general approaches to error prevention and detection as suggested in Chapters 3 and 4, in response to technical objective 2. These prevention and detection procedures emphasize the importance of man-machine dialogues in the control of input errors. The possibility of obtaining similar improvements via operator selection and training, or changes in procedures, work environment, etc. are also noted, particularly with regard to error types which have little if any possibility of control via improvement in the system dialogue. The suggestions for prevention and detection of errors reinforce the viability of alphanumeric keyboard data entry although other methods of data entry are considered. Since the recommendations are made without reference to a specific error problem or even a specific system requirement, there is no attempt to evaluate them beyond their potential impact on various types of errors. Each method of error prevention and detection will however have an impact on system costs as measured by both dollars and system performance criteria. To provide insight into these interactions the report calls attention to the potential of remediation methods designed to control one type of error to effect undesirable changes (for example, an increase in another error type and/or the timeliness of information disseminated). Therefore, the selection of any specific approach to control errors requires tradeoffs even when budget constraints are not a factor. Before implementing any alternative solution the report emphasizes and demonstrates the need to prioritize errors so that the solution does not, in effect, make matters worse.

Tradeoffs Between Benefits and Costs

The rational prescription for decision-making in the military environment, i.e., an evaluation of the tradeoffs between benefits and costs, is discussed in Chapter 5. In military applications, cost/benefit methodologies usually involve complicated mathematical and statistical procedures in an attempt to deal with the uncertainties, assumptions, and divergence in evidence and expert opinion which characterize the process. Chapter 5, therefore, presents a discussion in conventional terminology, of the pitfalls and shortcomings of the cost/benefit evaluation process and offers suggestions for the treatment of its "imperfect information."

Setting Priorities

Decision-making, choosing between competing alternatives, is the focus of technical objectives 3 and 4. The ultimate goal of improving the input interface in order to enhance system performance, embodied in Objective 4, carries with it an obligation to examine and evaluate every conceivable remediation alternative for every possible error -- an ineffective and inefficient approach at best, unless the decision-maker is completely unconstrained. Since error cost is measured by both economic and non-economic consequences, the decision maker will almost never enjoy the unconstrained position. Limitations on dollars available to remediate input errors (budget realities), minimum system performance requirements, constraints that are both external and internal to the ADP system, etc. all influence the choice of decision rule. Moreover, the domain of potential input errors is very large and since wide disparities exist when consideration is given to the costs of all possible errors, there is a need to reduce the remediation choice problem to manageable proportions. The aim of objective 3 is just that -- to prioritize input errors in order to provide direction for the decision-maker as to where to begin. Obviously, certain errors are much more costly in their effects than are other errors -- some errors, in a technical sense, may indeed be trivial in consequence. Thus, if the remediation decision process is to be rational and systematic, and resources are to be expended optimally, the satisfaction of objective 3 represents a critical prerequisite to objective 4.

The Multi-Attribute Utility Measurement (MAUM) Approach to Cost/Benefit Evaluation

Since the underlying task for both objectives is one of decision-making, it follows that the cost/benefit evaluative process is applicable to both choice problems. Conventional approaches to cost/benefit analyses usually offer copious detail on the wide variety of mathematical tools and formulations available to the analyst for his consideration of uncertainties and his manipulations of data. Usually, the more comprehensive the analysis, the more complicated the evaluation, with the result that findings become far removed from an intuitive base. Multi-Attribute Utility Measurement (MAUM), is an alternative decision-evaluation method that is psychologically meaningful, hence it makes an important contribution to decision-makers who are expected to render judgments that are intuitively reasonable. The essence of the MAUM procedure is flexibility in combining quantitative and qualitative evidence from different sources,

different lines of inquiry, and different techniques of investigation (including components of the more traditional cost/benefit approaches). Briefly, the ten step MAUM procedure is as follows:

- Step 1: *Identify the organization whose utilities are to be maximized.*
- Step 2: *Identify the issue or issues to which the utilities needed are relevant.*
- Step 3: *Identify the entities to be evaluated.*
- Step 4: *Identify the relevant dimensions of value.*
- Step 5: *Rank the dimensions in order of importance.*
- Step 6: *Rate dimensions in importance, preserving ratios.*
- Step 7: *Sum the importance weights, divide each by the sum, and multiply by 100.*
- Step 8: *Measure the location of the entity being evaluated on each dimension.*
- Step 9: *Calculate utilities for entities.*
- Step 10: *Decide.*

Application of MAUM to TOS Errors

Chapter 5 treats the MAUM procedure as a recipe for conducting the desired cost/benefit tradeoff evaluation whereas, chapter 6 provides an illustrative application of the procedure to the TOS input-error remediation problem. The two goals addressed in technical objectives 3 and 4 are treated as evaluation issues in the example of chapter 6. Thus, Issue #1 is error prioritization and Issue #2 is the selection of the "best" remediation alternative given a particular error or class of error. A schematic representation which helps support and clarify the logical flow of the decision-evaluation process is presented as a summary device (See Figure 2).

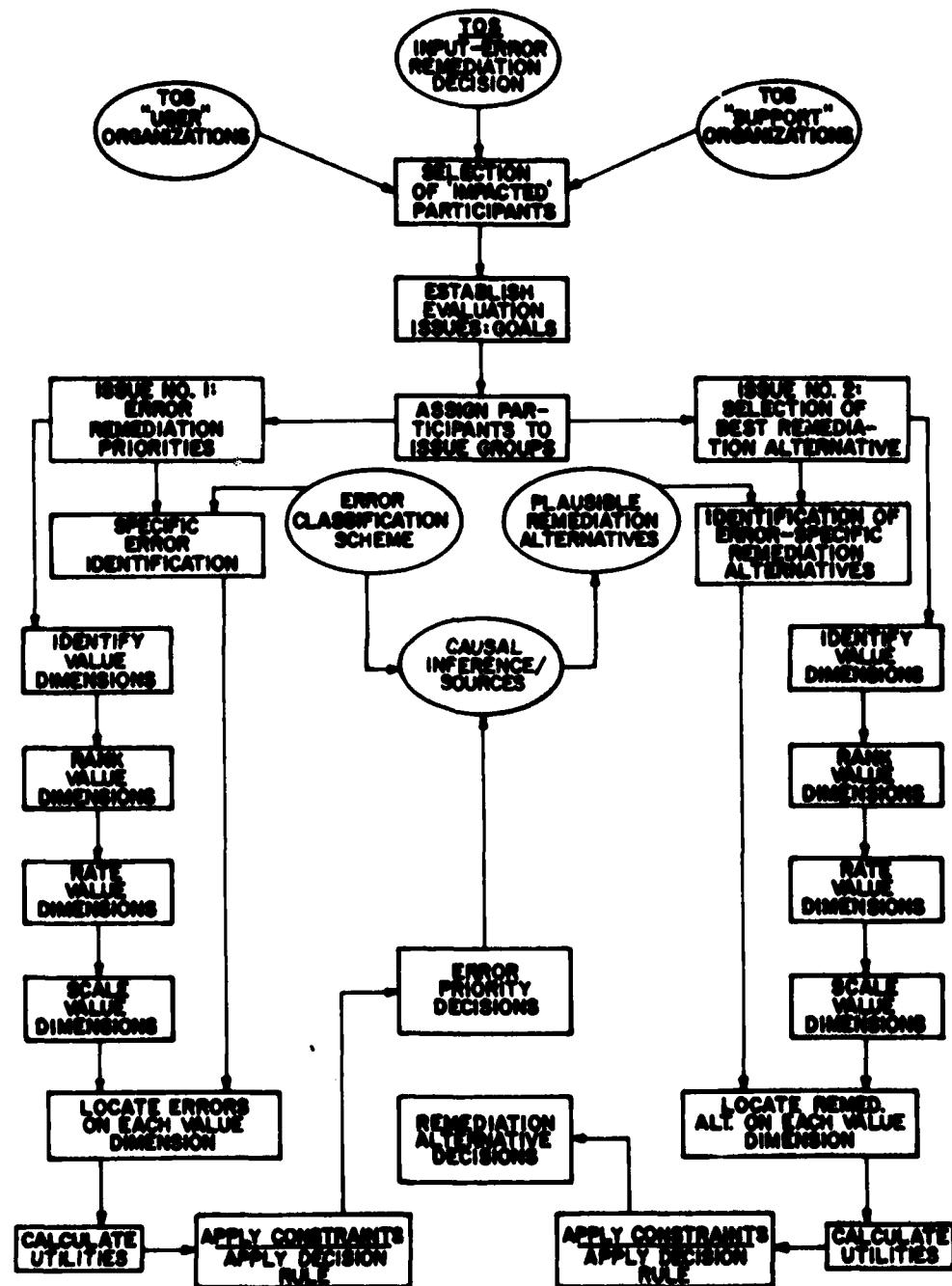


Figure 2. A Schematic Representation of the MAUM Decision Process Applied to the TOS Error Remediation Problem

I. INTRODUCTION

A. OVERVIEW AND PURPOSE

The scope and purpose of this document are much more modest than its title might suggest. It is only at the conceptual level, in the development of an error classification schema, that the broad scope of ADP operations in general is addressed. Beyond that, the viewpoint is specialized to attend, first, to the two primary input functional activities where human error is most prevalent, and second, to ADP systems in a military tactical environment. Within the boundaries established, the goal is to provide a conceptual framework for an analytic process which systematically relates error type to probable causes, suggest remediation/prevention alternatives and provide a cost/effectiveness based tradeoff procedure for selection of remediation/prevention approaches for new system design or existing system improvement.

The accomplishment of the project goal requires the integration of four basic components: (1) the error classification schema; (2) the constraints imposed upon the error remediation/prevention approaches (and system design, in general) by internal and external aspects of a system; (3) the possibilities for modification of the system interface, and; (4) an appropriate methodology for assessing the cost/effectiveness of various remediation alternatives in a complex, multi-dimensional context. Throughout the discussion, the attempt has been made to buttress the abstract and conceptual nature of the approach with examples drawn from actual tactical ADP systems with which the authors have worked. Specifically, the main focus is the U.S. Army Tactical Operations System (TOS) with other examples illustrative of the problems and approaches drawn from the USAF Tactical Information Processing and Interpretation system, Display, Control/Storage, and Retrieval segment (TIPI, DC/SR) and the USMC Marine Air Ground Intelligence System, Intelligence Analysis Center (MAGIS, IAC).

The following sections of this introductory chapter establish the rationale for a generalizable error classification schema, delimit the scope of this report to a concern with two major functional activities within this generalized system concept, and finally, define and describe each of the selected functional activities.

B. OPERATOR ACTIVITIES IN THE ADP INTERFACE

A universal schema for the analysis of input errors should be generalizable to most, if not all, ADP operations. To be truly comprehensive, the schema for error analysis should be general enough to deal appropriately with the specifics of existing systems, design alternatives for future systems and tradeoff possibilities involved in system remediation and upgrades. Our approach to the development of such a "system-free" generalizable model is to examine the variability of several existing systems to determine their inherent commonality as a basis for classification categories. Typically the examination of existing systems involves utilization of some form of job/task analysis (in either graphic or verbal form); a systematic, essentially chronological, view of the system process. This approach, because it is system dependent, maximizes our view of differences and totally beclouds any insight into commonalities. The very nature of the job/task analysis flow diagram depends on close correspondence to the specifics of the particular system being examined. What we require is a conceptual model of the basic activities which comprise any system in the general class of interest.

The diagram shown in Figure 3 is not a "flow" diagram; it is a schematic representation of the building blocks of the typical ADP system, be it communication, computation, report generation, or storage/retrieval. There are no arrows indicating direction of flow since this would require the introduction of various system specific assumptions, which, given our objectives, are inappropriate at this time. Beneath each of the transform blocks of Figure 3 are listed a group of functional activities which characterize man/computer interfaces. The seven activities are as follows.

1. Filtering: the preclusion of information from input to the computer, the data base or output to one or more specific users.
2. Data organization and code: the restructuring of source material into the records, format, and codes which the software is designed to recognize.
3. Electronic encoding: the transformation of information from one medium (typically hard copy or cognitive activity) into the electronic representation recognizable by the hardware.

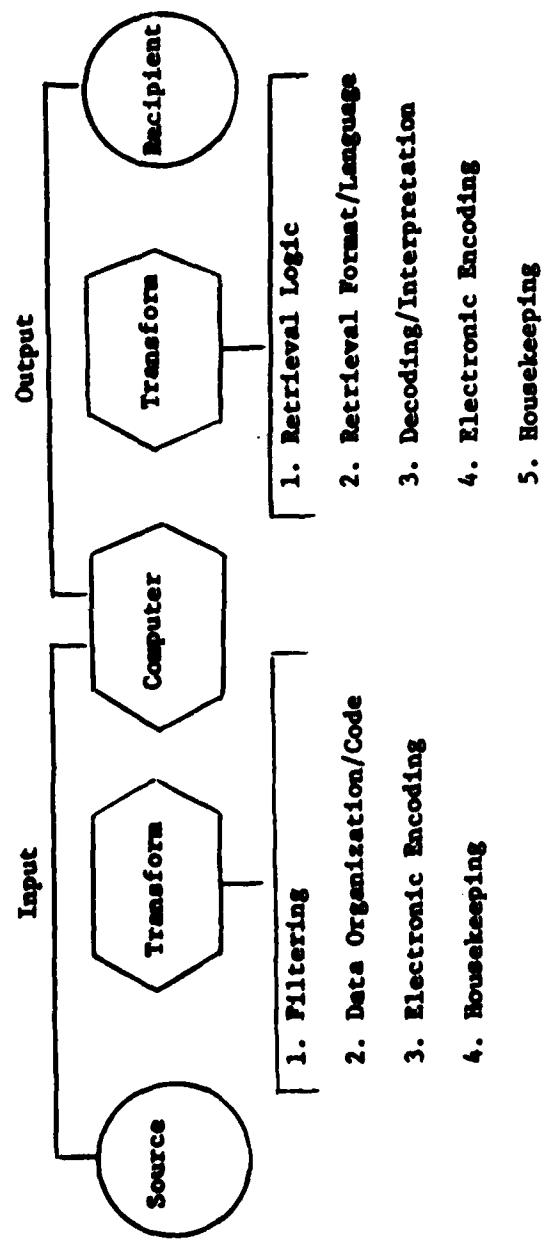


Figure 3. Schematic Representation of Information Systems

4. Retrieval logic: Analogous to filtering, this activity represents the selection of needed information from a data base which contains an overwhelming amount of data.
5. Retrieval format and language: this is the process by which a given retrieval strategy is organized and coded into software records, formats, and codes. It is to the retrieval of stored information what data organization and code is to the inputting of data for communication and/or storage.
6. Decoding/Interpretation: These activities, like the electronic encoding activity are analogous to a transducer. The information in whatever form it is output (coded/uncoded, graphical or list) must be interpreted in terms of the source scale and meaning. The quantitative and qualitative errors which are possible are similar to what happens in the parlor game where a circle of people secretively pass along a story too long or complex to memorize and are amused at the comparison of the initial and final versions. In the tactical situation such errors are, of course, not amusing.
7. Housekeeping: This activity refers to a broad range of tasks which must be accomplished if the I/O interface is to function as intended. They include for example, the required care of the workspace and materials; proper initialization of the system; I/O terminals, etc. maintenance of paper, film and tape devices, and the proper insertion of identification/header inputs.

The seven activities are not flow diagrammed in Figure 3 because their order of occurrence cannot be specified without reference to a specific system design. In fact, these activities may occur at several nodes of the system interface and be accomplished by one person, a person for each activity, or teams; and with or without computerized assists. Two activities (electronic encoding and housekeeping) are common to both input and output functions. Depending on the interface configuration, hardware, and system design, some of the tasks required within these activities may be identical for both input and output. The tasks of keying input and output requests are behaviorally identical (given that the same terminal is used for both functions). However, if input is from tape and output is via a terminal, the

encoding tasks for input and output are quite different. Therefore, in the design of systems these activities should be considered twice.

C. SCOPE

The man/computer interface has been described above as a collection of activities grouped according to whether they support input or output functions. A major problem in analytically isolating any one of these activities is that their level of independence varies as a function of system design, hardware, procedures, etc. In designing a new system, all of these activities need to be considered if errors are to be minimized. Where the goal is remediation of an existing system, all activities must be analyzed to determine those which offer the greatest opportunities for system improvements.

The analysis of input errors in this report will however, be limited to errors which are a direct result of either data organization/coding or electronic encoding. This focus is directed by technical objectives and resources of the contract under which this effort was undertaken.

The purpose of the data organization/coding activity (DO/C) is to transform information to be input to the computer from its source format to a format compatible with the system software. Transformations which occur prior to the system interface as a result of media conversions (e.g. voice to hard copy) are external to the system and therefore irrelevant to system evaluation since these conversions would occur with or without an automated system. Electronic Encoding (EE) is the activity directly concerned with transforming data from nonelectronic to electronic representation. Both activities can, in part, be accomplished or assisted using automated methods. For purposes of this report we are interested only in those tasks associated with the man/machine interface; therefore, for example, the task of keypunching is relevant while card reading is not.

The nature of these activities can be understood by comparing two different operational procedures: batch processing and on-line input. In batch process operations the data organization and coding is accomplished when data are entered onto card layout sheets. Spacing, formatting, and coding are accomplished before the data are submitted for keypunching, etc. Keypunch, key-to-tape, and verification are clearly separate tasks which accomplish electronic encoding. In on-line input operations, the data control cycle is compressed so that data organization/code activities often occur simultaneously with electronic

encoding. When messages are composed directly on a remote terminal there is generally no observable distinction between electronic encoding and data organization/coding. The user/operator does both at, what appears to be, the same time. For example, in the Army's Tactical Operation System, which has an on-line input capability, messages are sometimes composed on the CRT and sometimes composed as hard copy prior to data entry. The choice depends on the experience and preferences of the user and existing operating procedures at each station. Whether or not there is an observable separation between the two activities, the fact is that both activities necessarily occur and this has implications for error prevention and remediation. For example, misspelled words may occur because of a keying error or a cognitive error and redesigning of the keyboard will not eliminate the errors if the problem is one of learning which data codes are permissible. In other words, knowing the type of error does not guarantee knowing its cause when the two activities are confounded with respect to normal observation. The problem is further exacerbated by the fact that the DO/C and EE activities may also be confounded with other activities. For example, the omission of information from the data base may be a function of:

- Electronic encoding

ex: the operator skips a line when keying a printed text

- Data organization/code

ex: The operator does not know how to enter this information or perhaps assumes the system will obtain it from another source

- Filtering

ex: the operator intentionally skips the information because he considers it trivial, redundant, inaccurate, etc.

- Housekeeping

ex: switches set improperly so that data encoded are displayed temporarily, but are not retained in data base.

While this report focuses on the data organization/coding (DO/C) and electronic encoding activities (EE), the contribution of filtering and housekeeping activities to the production

of errors should not be underestimated. Too little filtering and the system becomes saturated, perhaps increasing the time it takes to input important data. Too much filtering and the accuracy of the output is affected by the absence of data inputs.

The potential impact of housekeeping tasks on input errors is even greater. At best, time devoted to housekeeping is a drain on manpower resources which can affect input effort. At worst, the complexity of housekeeping tasks contributes to mistakes or malfunctions which become the direct source of input errors. Because of the physical, mental, and emotional demands on interface personnel in a tactical environment and because of requirements for space and mobility, the housekeeping activity becomes a critical area where system improvements can significantly reduce system errors.

Equally important to the task of improving the utility of ADP systems, is the need to give similar considerations to each of the output interface activities. However, the purpose of this report is not to provide a guidebook for the design and troubleshooting of ADP system interfaces, but rather to provide the framework and procedures for the reduction of errors associated with only two of the four input activities: DO/C and EE. As such, it represents a step, however incomplete, toward the goal of improved system performance.

One other limitation should be noted before an error classification schema is introduced. While the schema proposed is applicable to all ADP systems, the development of the schema, sources of error and potential remediation has as a point of focus a more limited set of applications. First, we are primarily concerned with event based as opposed to real-time control applications.¹ Second, the discussions are more applicable to on-line teleprocessing applications than to batch processed jobs. Third, the recommendations in the paper generally assume that the input hardware includes a conventional keyboard and CRT display. Other devices such as light pens, joy sticks, or touch panels are only referred to where they appear to have a clear advantage. Finally, there is a deliberate bias in favor of tactical military systems in so far as serious constraints on selection, training, mobility, and security are assumed wherever appropriate.

¹Rouse, W. B. Design of man-computer interfaces for on-line interactive systems. Proceedings of the IEEE, 1975, 63, 847-857.

II. ERROR CLASSIFICATION SCHEMA

A. GENERAL

Before presenting the specifics of an error classification schema for ADP man/machine interfaces, it may be helpful to explore some of the general characteristics of classification schemas, particularly with respect to their design and utility. First, a classification schema can be defined as a method for categorizing observations into two or more dimensions which are hierarchically arranged. These dimensions, to borrow terminology from experimental design, may be completely crossed, fractionated or nested. Any combination of these arrangements is possible as well. Each dimension of the classification schema should, in effect, be a measurement scale and conform to the rules of scaling procedures. While classification schemas may utilize higher order measurement scales with which mathematical operations may be performed, the classification schema is most useful in applications where only nominative measurement is possible. Where higher order measurement exists, mathematically derived models can be developed from the available data. However, when only nominative scaling is available, the classification schema represents a substitute to quantitative methods to accomplish some of the same purposes for which mathematical models are used. In general these purposes can be summarized as the ability to organize, explain, or predict empirical phenomena.

Nominative classes within rudimentary classification systems must be exhaustive and mutually exclusive. Every object must be classifiable into one and only one class. Only statements of nonequivalence may be made concerning members of different classes. Unlike classes based upon higher levels of measurement, members of classes using nominative measurement cannot be compared with mathematical manipulations.

It should also be observed that classification systems are not unique. Any object can be measured in a variety of ways depending on which dimensions of the object are observed and the degree of resolution obtained. For our purposes it is essential that errors be classified in a way which aids identification of the cause of the error and ultimately its remediation. To the extent that the proposed classification system achieves this, it is successful, and the fact that other schemas might have been proposed is irrelevant.

To develop a classification schema for input errors we must begin by defining what an input error is (i.e. what is it that

the schema is to classify?). The specification of this "unit of classification", in effect, presupposes a preclassification or predefined population to be subclassified. We define an ADP input error as an unintentional discrepancy between truth (what is fact) and appearance (what the system purports to be fact) causally attributed to the man/machine input interface. To determine the causal relationship with the input interface, the source of truth for measuring these discrepancies must be some form of source document. Errors which occur because the source documents are wrong are not input errors. Discrepancies which occur from efforts to correct source documents are also not input errors because they are not unintentional.

Two populations of such discrepancies exist and these form the first dimension of the proposed classification schema. Discrepancies between information in the system and source messages may occur because the source message was not input correctly (errors of commission), or because the source message was not input at all (errors of omission). Omissions and commissions appear to be both a necessary and a useful classification dimension. They are necessary in that they define the populations of units to be classified and they are useful in that they relate, in part, to different preventive and remediation procedures. In the following sections, each type is subclassified and discussed separately. Figure 4 displays the subclassification categories for errors of omission. A similar figure is provided in the following section for errors of commission.

B. ERRORS OF OMISSION

With regard to the source document from which input errors are measured, three types of omissions can occur (Refer to Figure 4.) These types of omissions relate to the level of information omitted. Information omitted may be a single data element, a group of associated data elements, or an entire message set. Examples of each are as follows:

A data element is the smallest labeled unit of information to which a system can refer; e.g., in TOS the downgrading code which is one part of the security field, or the Unit ID are data elements. Similarly items such as Unit name, status of an installation, a country code or a UTM coordinate are elements.

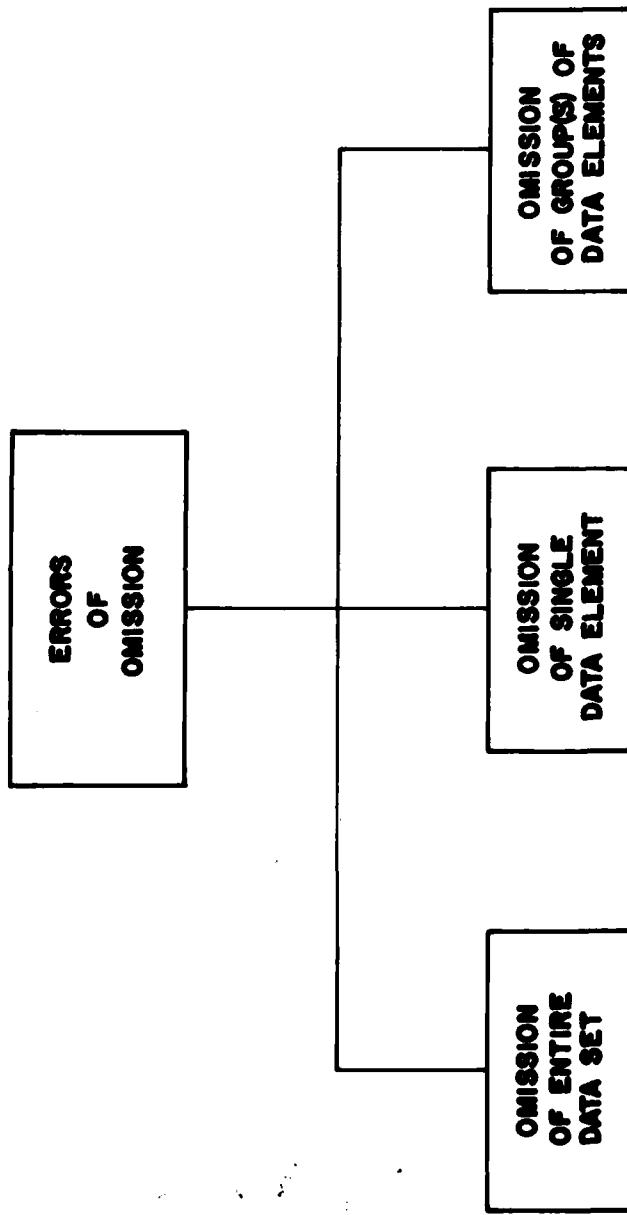


Figure 4. Error Classification Schema for Errors of Omission

A group of data elements are items of information which are associated by the system; e.g., in TOS a unit and its location is an example of an element group. In other instances the name, rank, and duty position of an officer, or the length, width, and orientation of a runway are groups.

A message set is a collection of similar groups of elements which, depending on the design of the software may be collectively input in a single format, record, etc.; e.g. in TOS a list of unit names and locations, or in other systems all of the information available on a unit or installation.

The omission of a message set occurs when all groups of elements of a common type contained in the source document are omitted. If at least one group of data elements are input and others are omitted, the omission is classified as a data element group. In the first case, the evidence suggests the operator does not recognize or know how to input this type of data, while in the second case his knowledge of how to input this type of data is demonstrated.

While the omission of a data set usually implies that some of the data contained in a source message were completely omitted from the ADP system, occasionally a data set omission can occur when all of the data have been input but not into all of the required locations or files. Some systems require that the same information be input twice. For example, unit locations may be maintained in a file of unit activity as well as in a file of logistical data; or a commander's name and personnel data may be in a unit record and in a biographical file. If the source document reported a change in the location of a unit, an input to each file would be required. An interrogation report identifying a new personality would require inputs to both unit and biographic files. A more complex variant of this is when one piece of information in the source document is associated with different additional data elements when input into two separate files. For example, unit name and location with left and right boundaries in a unit tactical disposition file and the same unit name and location with fuel and ammunition data in a logistics file (unit status) needs to be input twice, once with each of the two other elements of data.

The distinction between a data element group and a data set is independent of the input format or type of input dialogue and may in fact be identical in the case where a set is composed of

a single group. The basic distinction is that a set can contain repeats of the same element groups with different values. For example, a data set for an air field contains information on runway length, width, and orientation; a one runway airstrip set has one group of values; for multi-runway facilities, there are repetitions of runway data within the set. A single format or screen may be used to input multiple data element groups, so it may be possible to input an entire data set consisting of many data element groups on one screen. In other systems it may take several screens using identical formats to input each group separately.

The remaining type of data omission is the data element. Unlike the other types of omissions, the omission of data elements is more amenable to correction through improvements in the man/machine dialogue. The omission of data elements is also far easier for the system to detect than the omission of data sets or data element groups. The presence of some elements in a group can be defined as the basis of a conditional requirement that other elements be completed.

Causes of Omission of Message Set

The omission of an entire message set as a result of DO/C may occur for either of two reasons:

- Lack of knowledge of user/operator
- Incompatibility between source document and input dialogue.

The user/operator may not know how to input this type of information either because he is unfamiliar with the subject area (e.g. it contains intelligence information and his experience is in operations), or he is unfamiliar with the necessary system formats or codes. Even more likely is the fact that information may be overlooked because the user does not know what to input. Such omissions are encouraged when the arrangement or location of information in the source document obscures the fact that multiple data sets exist. This is particularly likely when the same information must be input twice. Baker reported that, depending on the source, from 1 to 2.5 formatted messages were required to input the information contained in the average source document into TOS.¹ In some cases, these were multiple data sets

¹Baker, J. D. "Acorns in Flower Pots/Psychologists in the field": paper presented at the United States Army Human Factors Development Sixteenth Annual Conference, El Paso Texas, Oct. 1970.

and in other cases they were groups of elements each requiring a similar but separate input message. In either case, the point of this finding emphasizes that the process by which source information is generated is generally not cognizant or concerned with the ADP system input requirements. Therefore one of the more difficult tasks in DO/C can be the identification of data sets in the source documents and the necessary procedures and formats for inputting each. It should also be noted that the requirements of system testing often dictate that this task, which can be quite real in an operational environment, is not exercised in the test environment, since source documents are usually simplified for test purposes.

The omission of an entire data set may occur as a result of EE for either of two reasons:

- a preformatted input sheet is misplaced
- the dedicated user/operator (i.e. the operator responsible for both DO/C and EE) having mentally noted the requirement and organized the information, for one reason or another fails to complete the input.

Depending on work load, work space, and backlogs, etc., preformatted inputs may be lost or discarded in an effort to catch up. Even when the DO/C activity is simultaneous with EE at the input device, the user/operator may still forget to input a data set even though he has correctly processed the source document and determined the required input procedures, formats, and codes. Such an oversight is more likely to occur when the data set is small and/or of minor significance with respect to other information in the same source document.

Causes of Omission of Data Element Group

The omission of one or more data element groups may occur for one of two reasons:

- Incompatibility between source document and input dialogue (DO/C)
- Skip a line or lose a page (EE)

When an omission is not classified as a data set omission the possibility that the user does not know how to input the type of

data can be discarded. While there is some possibility that the operator lacks the necessary knowledge of the specific codes needed to input specific information in a single element group, this situation is more likely to result in an error of commission. In general, the cause of the omission of a data element group will be oversight. When the omission occurs during DO/C the oversight is most likely to result from either carelessness or the incompatibility between the source document and input dialogue. Where data elements are organized in the source document as required by the input format, such oversights are less likely to occur. When the omission occurs during EE the cause of the oversight may be that the operator either skips a line or loses a page.

Causes of Omissions of Data Element

The causes of omission of a single data element and the activities in which they occur may be summarized as follows:

- improper presumption of default values (DO/C)
- improper location (DO/C)
- loss of place in source or preformatted document (EE)
- loss of place in dialogue with system, e.g. cursor position (EE)

ADP systems can define default values in a number of ways. A popular approach used by Fortran and many data analysis programs is to substitute a fixed value (typically "0" or "9") for the missing element. Another option is to leave the element blank. Still another approach is to use the last value with which the element was defined. When under the impression that he is making proper use of the default coding, the user incorrectly omits a data element, the system will receive either the wrong code or no code depending on the default value.

Another cause of data element omission during DO/C is correlatively related to an error of commission. When a correct data element is inserted into the wrong location two errors occur: one of commission with respect to the location where it was input, and the other of omission with respect to its proper place. The omission of data elements as a result of this cause is most likely when input formats are unlabeled or where similar labels are used for adjacent fields.

When the omission occurs as a result of the EE activity it is likely the result of the operator losing his place either in the hard copy or the dialogue with the computer, e.g., the input screen. Data element omissions from this cause are also not likely with input techniques which simplify verification.

C. ERRORS OF COMMISSION

Nawrocki et al. defined errors of commission as "an entry provided where none is desired".¹ We have chosen a broader definition for errors of commission to include all incorrect entries whether they are incorrect because nothing was desired or they are incorrect because the information was wrong in some other way. Our definition of errors of commission is therefore the converse of errors of omission. The schema for classifying these types of errors is shown in Figure 5. None of the categories is equivalent to errors from undesired entries. Errors which represent undesired entries may occur in any of the classification categories shown in Figure 5. The typical undesired entry may well be one of the two types of location errors generating a corresponding omission in the location where the data should have been entered. However, undesired entries may also be derived from fabricated data or data which has no place in the particular file or system.

The distinction between valid and invalid code becomes the central concept in analyzing errors of commission and presume restrictive definitions with edit and validation routines which define the boundaries of validity. By definition, invalid codes are reversible errors; (i.e., errors which are caught by the system and returned to the operator for correction). The major cost of such errors is additional time spent by both operator and system and thereby a potential decrement in the timeliness of information.

Without restrictive definitions all codes are valid even though they are not necessarily correct. Erroneous information is accepted by the system which may confound information retrieval, distort retrieved information, or confuse the user and waste his time because of the incompatibility of the information retrieved with other data. The detection of invalid codes is therefore an error remediation procedure. It does not prevent

¹ Nawrocki, L. H., Strub, M. H., & Ross, M. C., "Error Categorization and Analysis in Man-Computer Communication Systems", IEEE Transactions on Reliability, Vol. R-22, No. 3, August 1973.

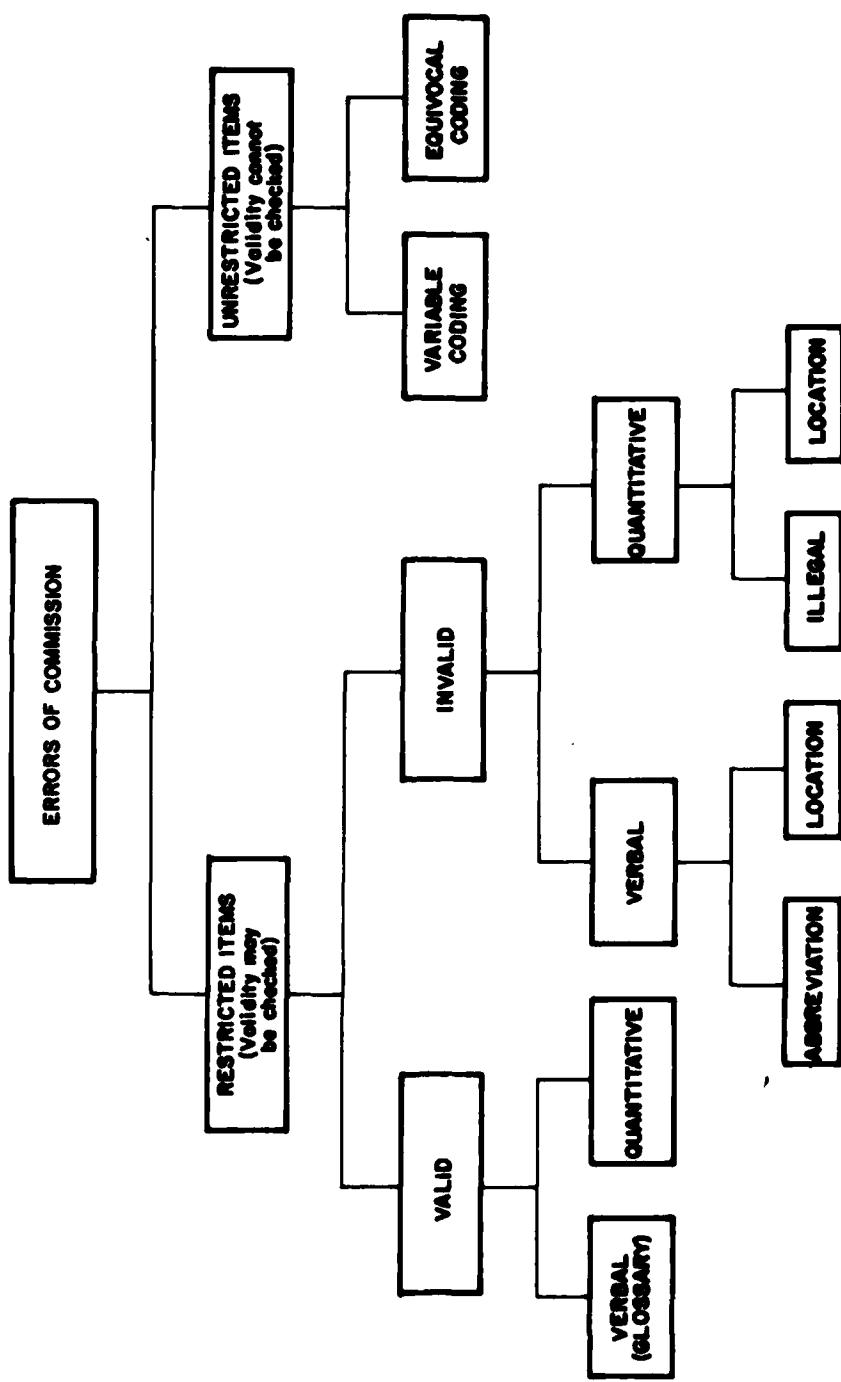


Figure 5. Error Classification Schema for Errors of Commission

errors, but instead detects errors before they produce more serious consequences.

Figure 6 provides a 2×2 categorization of all errors of commission. While there are other ways of categorizing these errors (and specific subtypes will be defined later) these categories are exhaustive and mutually exclusive. One dimension simply dichotomizes all inputs as correct or not correct (i.e., errors) with respect to any construct of truth available, while the other dimension dichotomizes inputs as to their acceptability by the system. The four cells labeled A through D can be rank ordered with respect to their desirability, based upon generalized consequences. In practice, systems are designed with restrictive definitions for some items and not others. Validity checks are more feasible when the list of valid codes is short and unconditional (i.e. definitions do not change as a function of other coded information). Validity checking can require extensive storage for representation of valid codes and in some cases more complex software to program heuristic procedures and conditional logic. Increasing the number of validity checks (assuming no change in input code) reduces the number of errors categorized by Cell D in Figure 6 and increases the number of errors categorized by Cell C. The procedure does not prevent the error, but provides instead a method for remediation. However, since errors in Cell C are also undesirable, attempts should be made to reduce them. The problem discussed in more detail below, is that some of the methods available to reduce errors in Cell C have the undesirable potential effect of increasing the number of errors in Cell D.

RESTRICTED ITEMS - VALID CODE

The first level of error classification shown in Figure 5 separates errors as to whether or not they involve a restricted or unrestricted item. Errors on restricted items may result in either valid or invalid entries while errors on unrestricted items are, by definition, valid. Both valid and invalid errors on restricted items may involve either verbal or quantitative codes. The list of valid codes for verbal items can be compiled a priori in a glossary. The list of valid quantitative codes on the other hand would normally be too lengthy to enumerate and instead they have their validity defined by some numerical expression, e.g., upper and lower boundaries. Incorrect but valid codes in verbal elements are referred to as glossary errors while incorrect but valid quantitative codes are quantitative errors.

Cell A This is the cell into which we would like all inputs to fall. The input is a true representation of reality and is accepted by the system.

Cell B This cell is shaded to indicate that it is least likely to occur. It is however, the next most desirable alternative in that correct inputs not accepted by the system suggest changes for system criteria which prevent such rejections from occurring in the future.

Cell C This cell is more desirable than Cell D because given that an error is made, it is preferable to have it detected by the system, permitting correction, than to have it accepted.

Cell D This is the least desirable cell because these erroneous inputs are accepted by the system and may interfere with subsequent retrieval or convey misinformation to its recipient.

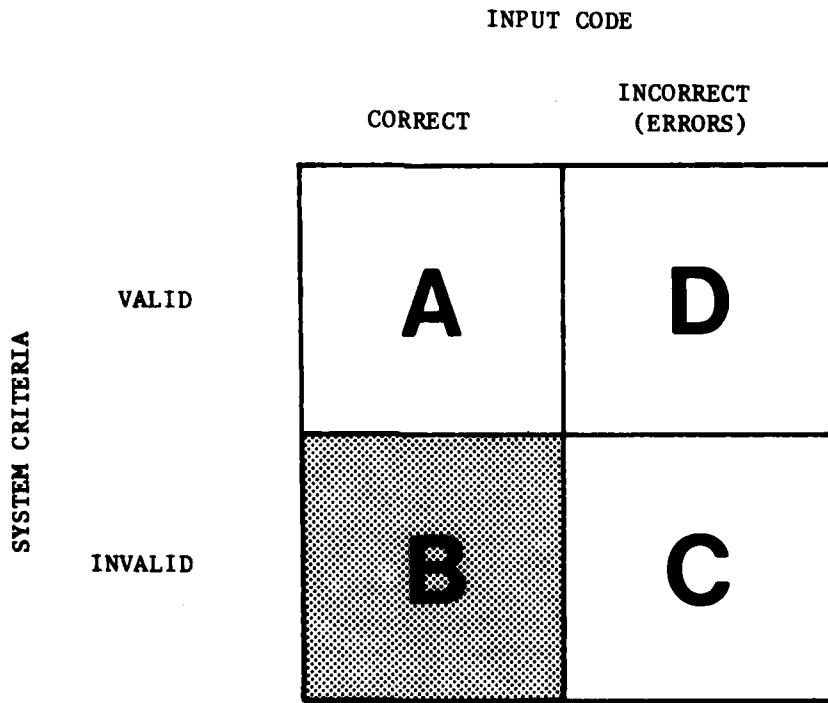


Figure 6 Two-Way Classification of Errors of Commission

Glossary Errors

Glossary errors occur in three basic ways. Either the user/operator can rely on his memory and recall a valid but incorrect code, refer to a list of restricted codes and make an error of recognition, or he can refer to the glossary and make an error in reading or transcribing what is there. For example, recall errors were commonplace in the USAF DC/SR where "C" numbers were used as retrieval qualifiers. Each file had a set of C numbers and the sets overlapped from file to file but the definitions did not; C12 in one file might mean "runway length" while in another file C12 meant UTM coordinate. The use of C12 in either file was valid but could be incorrect due to the varying definitions.

Even when abstract codes such as these are used consistently, the possibility of recall errors exists. Consider for example, the following coding systems for a selected set of man-made terrain descriptions:

<u>CODING SCHEMA</u>	
A	B
Crossing, river	MM2
Junction, railroad	MM3
Junction, road	MM4
Junction, trail	MM5

Using coding schema A, the user/operator may for one reason or another associate MM4 with railroad junction and MM3 with road junctions and should the user/operator refer to the glossary, the potential for the same error exists in that reading left to right he may become dislocated and transcribe the wrong code. Such errors of association and, in general, all recall errors can be reduced by either increasing the use of a glossary or by using abbreviations or mnemonics such as those represented in schema B above. Now the valid code for railroad junction has the embedded letters RR. Not only is a recall error less likely but when the user refers to the glossary, transcription errors are less likely because the opportunity exists for the alert user to catch his error if the mnemonic he is entering does not have an embedded RR. Although the use of mnemonics should also reduce recall errors, reversal errors such as TJCTR are possible. This, however, could be detected as an invalid code (see discussion for abbreviation errors).

Quantitative Errors

The causes of quantitative errors include the following:

- Metric conversion
- Decimal placement
- Rounding
- Careless transcription
- Character transposition

The best approach to eliminating measurement conversion errors is to reduce the need for conversion. The software should be programmed to accept quantitative inputs denominated in the unit of measurement contained in source documents and instrumentation. In situations where sources are highly variable with respect to unit of measurement, it would be preferable to have the software allow the user to specify the unit of measurement than to have the user make a conversion prior to entry. Specification by the user can be accomplished by inputting two data elements; one the quantity, the other the measurement unit. In the case of coordinates, one source (USAF) may use latitude and longitude (LAT/LONG), another (ground forces) universal transverse mercator (UTM). Data entry can be simplified and errors reduced if the system will accept both and automatically enter the conversion value when either type is input.

Errors in decimal placement may occur either overtly from misplacement of the decimal or covertly from use of an improper multiplier. For example, the user/operator may input the number 6000 to designate a distance of six thousand meters, but because the field is defined as kilometers the input is interpreted as six million meters. Two possible solutions to reducing this type of problem are either to display the scale (i.e., kilometers instead of meters) adjacent to the input field or to allow the user to input the measurement scale as suggested for conversion errors above. Still another approach applicable with covert errors of metric measures is to always define the input in terms of meters or kilograms and allow the user to insert a decimal wherever it is needed.

Should rounding errors occur and represent a significant problem they are easily eliminated by extending the length of the input element to allow for input of one more decimal place than the desired level of accuracy.

Careless transcription and character transposition are the most difficult to explain. They may be caused by the environment, fatigue, boredom, personal behavior, etc. Where the cause of carelessness cannot be removed by changing the environment, work load, etc., some form of verification of data input will be required. Obviously, having a second person duplicate all input activity is a costly procedure justified for only the most error prone information elements and where errors have the most serious consequences. The most practical solution for most information elements and most systems is to train people to check numbers more carefully than other inputs since quantitative data are most difficult to check. Interactive dialogues are also useful in helping the user verify his own inputs.

RESTRICTED ITEMS - INVALID CODES

Invalid codes refer to errors detectable by edit and validation routines in the software. As stated earlier, the purpose of validity checking is to reduce the amount of incorrect data accepted by the system. Invalid codes are less costly in that they do not result in the spread of misinformation. Their cost is reflected in wasted time and the consequences of wasted time which may be significant. In the sense that invalid codes prevent erroneous information from being accepted by the system, the invalid error (Cell C, Figure 6) represents a reduction in valid errors (Cell D, Figure 6). As noted earlier, attempts to reduce the number of invalid errors are useful as long as the method of reduction does not result in erroneous information being accepted by the system.

Verbal Codes

The typical method of using edit and validation is to check against a closed-ended, a priori list of valid codes. TOS provides an extensive glossary of valid codes for a large number of elements. The application of validity checking to nominative elements may be absolute or conditional with the latter occurring when the list of valid codes is changed as a function of other entries either in the same message or elsewhere in the system. For example, a list of valid equipment codes might change as a function of unit type; or valid message originator numbers might change as a function of message type and inputs of the system controller.

Verbal coding errors can also be of two types: errors involving abbreviations and errors involving location. Errors resulting from use of improper abbreviations are difficult to

discriminate from typographic errors. Nawrocki et al. used frequency criteria assuming that typographic errors were random and therefore not likely to be repeated.¹ For our purpose, abbreviation errors include all errors which result from the use of an illegal group of characters representing a verbal code; whether such code is an English word abbreviation, mnemonic, or an alphanumeric nonsense symbol. These errors, if not purely mechanical, can occur for one of several reasons which are primarily related to problems with memory and motivation. Memory failure is the most obvious contribution to these types of errors. Codes used less frequently are most easily forgotten (if, in fact, they were ever learned) and to save time, the user tries to recall the code. When the codes used are abbreviations or mnemonics, the tendency is probably greater to try recollection as opposed to consulting a glossary. The less motivated the operator, the more likely he is not to know the code and the less likely he is to look it up.

The other type of error of commission involving the use of invalid verbal codes is referred to as a "location" error. Our interpretation of this error type is consistent with Nawrocki et al. except as noted above, we would include here some of what Nawrocki called errors of commission. A location error therefore is one which is detectable because the code input is invalid (whether or not anything is desired or is irrelevant); however, the issue which remediation must address is that the code selected would have been valid if input in the proper location.

The major causes of this type of error are either carelessness or failure of the format to clarify what is needed. Such confusion may arise particularly with tabular formats where labels are necessarily brief and the coding packed closely together.

Quantitative Codes

The last type of invalidity error which can be detected are quantitative elements. These include elements such as time, coordinates, and quantities. As was the case with verbal data elements, quantitative data elements may be invalid either because they have been input in the wrong location or because they contain an alphanumeric expression which is defined illegal in either an absolute or conditional sense. The causes and prevention of location errors in the quantitative elements are identical to those already discussed for verbal elements while the causes and prevention of illegal quantitative entries are the same as those discussed earlier for valid quantitative data elements.

¹Nawrocki et al. "Error Categorization".

UNRESTRICTED ITEMS

By definition, all codes for unrestricted items are valid and therefore all errors made are accepted by the system. There are two types of codes in unrestricted items which produce errors. First there are equivocal codes (i.e., a code used on different occasions to mean different things) and second there are variable codes (i.e. a code used with a meaning for which another code was already used). Both types of codes create input errors. The first type results in the spread of misinformation while the second type of error reduces the amount of information retrieved. There are two major causes of equivocation among unrestricted codes; either the coding conventions are inadequate to guarantee uniqueness or the data element length is insufficient to develop a unique code. Variable coding is also the result of inadequate coding conventions, including rules for spacing and punctuation.

While some coding conventions can be implemented to regulate coding of certain elements such as unit name, the potential variability for coding place names and installation names is enormous and beyond solution through coding conventions alone. What, in effect is needed, is an open-ended glossary, one which can be used to standardize previous entries and be expanded as the need occurs for new entries. The software could be designed to check the input code against the open-ended glossary for that field and if a match is not found, display the list of previously encoded items so the user/operator could determine whether a code already established is equivalent in meaning to the name he wishes to input. Aside from coding conventions and computerized assists of recall, errors of variable coding will also occur as a result of misspellings from typographic or other transcription mistakes.

For example, in the case of place/installation names, the name may vary but the location will not. By linking, in the glossary, name and coordinates at the time of initial entry, multiple names (which confuse retrieval) could be caught and eliminated. If this approach were to be implemented, it would, of course, be necessary to allow for certain types of multiple names for a given location. For instance, "Andrews GCA radar site", "Andrews AFB", and "Suitland, MD" all exist at roughly the same coordinates. Similarly, several different types of defensive fortifications can be expected to exist at the same general coordinates of an enemy strong point.

III. SYSTEM CONTEXT FOR ERROR REMEDIATION

A. GENERAL

The prevention of input errors at the man/machine interface of military ADP systems may be accomplished by the manipulation of one or more characteristics of the interface; the internal system factors. The utility of any design recommendation in the reduction of errors depends on both its inherent effectiveness and its cost feasibility. The cost of any change in the interface design depends on the nature of the change, the stage of system development, and on the existing system design characteristics. The ultimate effectiveness is determined and the trade-off among proposed changes is constrained by the external factors of system requirements or objectives and the operating environment. Depending on the system design characteristics, a change to any design feature may be interactive, affecting costs in other areas, internal and external, which make the change impractical. For this reason, it is not possible to draw an unequivocal relationship between error type, cause, and prevention remediation technique. The prevention/remediation of human input errors must therefore be addressed within a more general construct of system design.

The process of system design, whether considered from the beginning of a new design effort, undertaken to satisfy a newly stated operational requirement or observed in the process of modifying an existing design to improve performance, is an effort to solve a problem which exists in satisfying a specific objective(s) in a particular environment. The system is the solution, enabling the accomplishment of the objective in a new and better way, enhancing the capability of the organization to do the required job within and in spite of the constraints on performance imposed by the environment. In traditional system parlance, the objective and the environment are "external"; the system hardware, software, procedures, and personnel are "internal". In our concern for identification and selection of remediation alternatives for the prevention or remediation of human input errors, the focus is upon tradeoffs among internal factors. The effort is, however, doomed to failure, if that tradeoff process ignores the external factors and their influence upon what alternatives are feasible. It is the purpose of this section to consider the constraints imposed by the military mission and the tactical environment upon the solution space available to the system designer, and which must ultimately be translated into restrictions upon the selection of remediation alternatives.

Perhaps the first and most basic point to make is, that while the internal factors involved in military systems are closely akin to their commercial, civilian counterparts, the external factors are dramatically different. The job to be done, the mission to be accomplished, the environment in which the system and all its components must exist and perform, do not have civilian counterparts. To be sure, both military and civilian systems can be characterized as management information systems. The basic internal factors are the same and the same components may even be identical; but, just as an automobile and a main battle tank are both vehicles and built of the same raw material, the job to be done and the conditions of performance dictate quite different solution configurations. The special demands of the battlefield levy compelling requirements for a whole series of special system "abilities": survivability, supportability, transportability, interoperability, reliability, maintainability, and availability. These requirements, in turn, have their impact on system design in the form of constraints or limitations imposed, in great detail, in the system specification. There is, unfortunately, no generalizable, systematic way of dealing with these requirements since there is no inherent priority ordering, universal definition nor intrinsic interrelationship among them. Different levels of each are set for different systems; varying compromises among them are struck from time to time, and; the selection of system components must be adjusted accordingly. The following is, therefore, an examination, in arbitrary order, of a variety of the attributes and manifestations of these constraints, as observed in a variety of tactical military systems upon the selection of the mix of internal system factors. It is the intent to demonstrate the range, complexity, and subtlety of these efforts upon the initial component selection and upon the tradeoff process involving input error remediation alternatives.

B. EXTERNAL INFLUENCES

One logical starting place is to begin with the requirement for mobility (or at least transportability) since this traditionally vital determiner of combat success has assumed even greater importance in modern tactical doctrine. The most obvious consequences of the mobility requirement involve size and weight. The system must be packaged within a certain limited envelope and must not exceed a certain weight per unit. Whether the system is to be packaged in towable, air transportable shelters, carried in smaller shelters on 5T 6 x 6 trucks or dependent upon remote terminals carried in command tracks, the overall envelope size and package gross weight cannot be waivered to accommodate larger

or heavier components, no matter how desirable they may be from a systems performance point of view. A shelter trailer larger than 8 ft x 8 ft x 20 ft cannot be air lifted by a C-130 and while larger airplanes are available, there are neither enough of them nor can they themselves be accommodated in a truly tactical environment. Gross weight is likewise limited by the safe lifting capacity of cargo type helicopters or the all-terrain payload of the 5T 6 x 6 truck. Such considerations even filter down to the individual component level where there are stipulations on the weight of items which must be relocated within the shelter for transport or maintenance.

Another aspect of the mobility requirement which, while not as obvious as the size/weight issue, is just as pervasive; perhaps even more so since it acts counter to the design goals of decreased size/weight: the requirements for ruggedization. To withstand the rigors of mobility testing, which requires dropping, railroad humping, miles of pounding over Belgian block roads and similar tortures, equipment must be shock mounted, potted, reinforced, and structurally strengthened. This, of course, leads directly to increased size and weight/unit and results in reduced system size and capability since for a given component a substantial portion of the allotted space is taken up in mountings, braces, etc. Furthermore, there is the additional consequence of a built-in technology lag. The process of modifying and testing a specific new component (e.g., disc drive, printer, CRT display, etc.) to qualify it under military standards for inclusion in a tactical system is expensive, demanding, and time consuming. Newer technological advances in I/O devices, CPU's, etc. readily available to commercial installations (and government/military fixed facilities) may require as much as three or four years of additional engineering/testing to qualify and become available to the tactical system.

In the same vein, considerations of survivability and vulnerability influence component selection/package in much the same way. The necessary provisions to protect equipment from extremes of temperature, dust, humidity and various forms of corrosive agents consume space and add weight. So too do the filters and shielding necessary to guard against outside electronic interference and to prevent the emitting of compromising emanations. Engineering and packaging electronic components to survive temperatures ranging from -60°F to 130°F, dust storms, weeks of 100% humidity at tropical temperatures, prolonged exposure to salt, fog, and similar climatological extremes is not an easy task and the testing itself can be destructive if the designs are not 100% successful. Protection in the electronic environment of the

battlefield is difficult. The range of emitter characteristics from which the system must be shielded is vast and the probable proximity and power output of the emitters makes the requirement much more stringent than needs to be considered in the civilian community. As for the emanations side of the electronic concern, the problem is at least as critical, since the enemy can be presumed to have both the motivation and the technology to attempt to collect and decipher any coherent signal from the tactical ADP system. Failing that, the mere detection of an identifiable electronic signature on the battlefield is all that is required for accurate locating and targeting. For both interference and emissions, then, components must satisfy rigorous and unrelaxable standards which far exceed the original civilian design specifications thus interposing a further lag in application of the most modern technology in the tactical environment.

Supportability requirements stemming from considerations of the tactical environment also constrain many of the system designers alternatives. The variety of ways that the supportability requirement is manifested is surprisingly diverse. The most commonly thought of dimensions involve supplies and spares. These are certainly important considerations. There must be paper for printers, acetate or vellum for plotters, maps, reference materials, message forms, and the like. Replacement parts must be readily available to the maintainer; it is of small use to be able to quickly fault isolate to a line replaceable unit if the required LRU is not on hand wherever the system may be deployed. There are numerous other concerns, as well. These include: special test equipment, power generation equipment, special handling equipment (e.g. hoists, dollies, etc.) and trained maintenance personnel. Introduction of new system components can, if not rigorously controlled, lead to greatly increased costs in spares provisioning, special equipments, and manpower requirements and, in a tactical environment, even if the dollar costs can be supported, the physical requirements may be prohibitive and the trained manpower may not be obtainable. These supportability concerns, while true of any "system" (including weapons) become even more stringent in this context. Tactical ADP systems are ancillary, not primary, to the major ground force mission. Even when fully developed, tested and deployed, there will be relatively few copies of any given system. For example, the planned deployment of the Intelligence Analysis Center (IAC) being developed by the USMC calls for three production copies, total. The USAF will probably never procure more than half a dozen Display Control/Storage Retrieval (DC/SR) segments. The ultimate Army requirement for TOS will be considerably greater but still, in terms of number of copies, small, relative to other

types of systems. When translated into terms of allocation of supporting resources, tactical ADP systems will inevitably have lower priorities which in turn means that the systems must be smaller, simpler, and must employ components in common with other systems (e.g. use of a common micro-processor in all systems requiring such a component).

Interoperability is another system attribute demanded by the special nature of the tactical environment; an intelligence or operations system isolated, cut off from direct communication with other elements and echelons of a field command, is essentially useless. This requirement has serious design implications for all aspects of internal system factors; hardware is only the most obvious. It may be necessary to have two different tape drives (7 and 9 track) for example, because other systems already in the inventory use one or the other. Communication lines with different cryptomodems are required to interface with different elements of the command. Radios with different frequency ranges are required for tie-in to nets at various levels. The impacts on software can be just as great. For example, the current work on Joint Interoperability for Tactical Command and Control Systems (JINTACCS) has as one of its goals the development of standardized man/machine readable message formats. Once developed, tested, and accepted, the possibility of electronic processing and automatic data base update become very real. All that will be required to realize the potential benefits in speed/accuracy of information handling will be the development of system software to accomplish it. Such software while clearly within the state of the art is sure to involve large, complex programs; a certain burden on already memory limited tactical systems. Until the implementation of such programs (and even afterwards for many applications) it will still be necessary for the personnel and procedures components of the system to cope with formats which, in order to accomodate the computer, are less than ideally structured for "manual" processing. (An excellent example of this is found in the recent change of the IPIR/SUPIR format in DIAM 57-5). The important point for this discussion is that procedural change within a given system will be seriously constrained by requirements imposed to foster interoperability of various systems.

Reliability, maintainability, and availability are often spoken of as a group (the RAM requirements) and while each has its own definition and specific concerns, each depends on the other and their joint effect is the concern here. In the austere and time critical environment of the military system, it is obviously essential that the hardware components do not fail, that when they

do, the failure can be quickly isolated and repaired and that the system is in an operating state a high percentage of the time. For the system designer these concerns are reflected first in hardware component selection but other system factors are also involved: diagnostic software must be provided; maintenance personnel must be assigned; operating procedures must allow time for regular preventative maintenance. If the specified reliability level cannot be met on a single path, then alternative paths (backup components) and associated reconfiguration software and procedures must be provided. Maintenance activity must be facilitated by special fasteners, slideout/swingout racks and similar special features. Finally, and of ultimate importance to this discussion, it must be recognized that the job to be done by the system, the mission to be accomplished must be done even in the event of catastrophic failure of the computer. It is clear that this requirement represents a serious constraint on system design and the allocation of tasks between men and machines. There must be incorporated, in the system, provisions for degraded mode operation and for orderly transition to normal operations following a period of manual or less than full capability operation. Such provisions involve all major system factors. System personnel must know how to do their jobs with and without ADP support; procedures must require maintenance of a man readable record of essential data on a periodic basis; the software must facilitate the production of this hardcopy backup material and the necessary I/O hardware must be provided. In the event of loss of the main computer for more than the specified mean time to repair (MTTR) and assuming the communication capability is not lost, the operation can be maintained in a manual mode until the necessary repair/replacement can be effected.

Having considered some of the major influences of the military mission and the immediate tactical environment, it must be noted that the list of external factors constraining the solution space of the system is not complete. Broader aspects of the military situation are also involved; aspects which are legitimately included in the major externals of objectives and environment but which are not necessarily as directly discernible in the system specifications. Included here are such generic concerns as personnel policy, budget, and user acceptance.

The label "personnel policy" is employed here to distinguish this external dimension from the internal personnel factor and to emphasize the far reaching implications which force the system designer to compensate for rather than fully utilize the personnel component. The problem here is more than the widely recognized deficiencies of an all volunteer force. It involves such factors as assignment/rotation policies, unit TO's, primary/secondary

MOS's and the difficulties of maintaining combat readiness in a peace time military. The basic point involved in all these problem dimensions is that they arise not because of any error, deficiency or poor policy decision, but rather as a consequence of attempting to optimize a "system" much larger than the tactical ADP system -- a system of which the tactical system is a small subset: the entire military establishment. In short, problems arise for the tactical ADP system designer not because there is something basically wrong with military personnel policy but because of the priority given to such ancillary systems relative to goals and objectives of the armed forces as a whole. Regardless of the reasons, the constraints placed upon the selection/utilization of the personnel component in tactical ADP systems are real and severely limit the design alternatives as well as remediation alternatives. Any attempt to design a system or to improve the performance of an existing system which depends merely on specifying operator/users who are more intelligent, better motivated, more thoroughly trained in both the system operation and the information content area being served, who are available in larger numbers, stay with the job longer, have no additional outside duties and can practice daily, is sure to be futile.

Furthermore, the question is not how or whether such an ideal state can be achieved but how to design or modify the tactical ADP system to best utilize the personnel available. Design alternatives/remediation alternatives must be limited to those which are feasible within the larger context, anticipating that operator/user knowledge and capability entry levels will be limited, that training time will be restricted, that the unit to which the system is assigned will have a TO which cannot be significantly expanded and a personnel complement which will be less than full strength. Many of the system operating personnel, particularly at lower echelons, will have other duties and responsibilities some of which will take precedence over system operation.

The external influences of budget and user acceptance form a kind of conceptual bridge between the foregoing discussion of constraints on alternative selection and the cost/effectiveness approach to the tradeoff among alternatives. Budget concerns are obviously accounted for on the cost side of the evaluation and user acceptance is intimately involved in the effectiveness considerations. Both, however, have a prior and independent reality which can be employed before any attempt is made to reduce a specific set of alternatives to dollars and performance improvement scores. Certain alternatives (for design or

remediation) can be eliminated out-of-hand as too costly; allocations for the system will simply not support certain alternatives no matter how advantageous to performance they might be. Other alternatives may be eliminated simply because of the lack of user acceptance. An example of this latter case can be found in the USMC IAC where an automatic update capability for certain types of data was rejected by commanders, even though it would completely eliminate operator input errors and substantially increase timeliness. Their reason was simple: nothing should go into the data base without review by their command. In other instances, automatic purge criteria are overridden, in spite of their obvious benefit to efficient data base management, because the command users lack confidence that the procedure will not throw away good data along with the out-dated. It is important to recognize that both budget considerations and the degree of user acceptance/confidence are dimensions of the environment in which the system is designed or modified. They operate to constrain the solution space in the same way as the physical dimension of the tactical environment.

C. INTERNAL INFLUENCES

The characteristics of the specific operator personnel designated for the system are an important source of constraints on the selection of system design alternatives. The design of an input dialogue, for example, must give close consideration to the characteristics of the person inputting the data. Characteristics which seem most critical to the design of a dialogue may be classified as either psychological or skill factors. Psychological factors which may impact on error rates include ability, boredom, and motivation. Skill factors include basic aptitudes and intelligence as well as experience with the subject matter of the system and with ADP in general.

While the input dialogue must be suited to the psychological and skill characteristics of the operator, these characteristics may be changed when possible to provide greater flexibility to the options for the design of a dialogue. Such changes are typically achieved through selection and training. Selection being most applicable to acquiring operators with the desirable psychological traits and skills in the subject matter, while training is more applicable to the acquisition of operators with skills in ADP and a particular system dialogue. If, however, because of other constraints, the system dialogue cannot be optimized for ease of use, consideration should be given to the selection of operators with an ADP background. This, however, may pose a dilemma where the system is equally in need of

operators with very specialized subject matter expertise. In this situation the population of operators with both ADP and subject matter expertise may be too small to provide the support required.

With respect to training, we would emphasize the importance of OJT. Many of the errors identified in system tests would undoubtedly go away once the system was operational, and the operator gained additional experience. This is not to give license to poor design which relies on the adaptability of the human to make the interface function. Given a system still in the design stage, it is obviously worthwhile to try to optimize the software design with respect to user abilities to minimize all types of errors. However, much of the effort in system redesign (which includes both software and hardware) seems to result from the desire to play with "the toy" than to make a serious effort to get a job done. Some systems obviously need redesigning, either because they were poorly designed to begin with or time has passed them by. Our argument is not with redesign or system modification as a legitimate system development activity, but with the trend toward the institutionalization of this activity in the design process.

The major problem in giving operators enough training can be a Catch 22. Given a system which is not operational or one which is only deployed a few times a year (e.g. in tactical exercises), time devoted to training is often above and beyond the trainees normal duties. Operators who can be spared for training are generally least competent in the subject matter area while those who are competent do not have the needed time. Those who are competent to perform the job could be made available for training, however, if the system were operational and therefore helped them accomplish their job while they were training. However, before the system can be helpful its effectiveness and accuracy must be demonstrated which cannot be accomplished before the operators are trained, i.e. CATCH 22.

The most practical solution to this problem is to modify system requirements, narrow the system's scope and integrate the system into the user's daily activities. More highly trained operators might be obtained if the system were first designed to accomplish a finite number of objectives with new capabilities added after system acceptance is achieved. Further, system acceptance can be facilitated if the system is designed to be used on a daily basis. This latter suggestion is relevant only to systems whose major utility is in special situations, e.g. tactical exercises, crisis situation, etc. TOS for example, is

designed for the tactical environment, however, much of the data contained in TOS files could be used for other purposes on a daily basis, e.g. enemy history data, authorized assets, etc. The system used on a daily basis might be a modification of the one used in special situations. If, however, the same design concepts are used, training will be transferable.

When the constraints introduced by operator skills and psychological characteristics cannot be overcome through selection and training, restrictions are imposed on the design of procedures. A key area of operating procedures which must be considered is the work/task allocation. Alternative work/task allocation procedures can be considered as either horizontal, vertical, or temporal distributions of work. In a horizontal distribution, the procedures would allow for specialization, i.e., different operators for different areas of subject matter, files, types of input, etc. In TOS, for example, there is clearly a need for specialized operators in intelligence and operations. Specialization may only be necessary or possible at some input stations, levels of operation, etc. At lower echelons, for example, personnel resources and work space may make distribution of the work load among specialized operators impractical.

A vertical distribution of work refers to the allocation of ADP activities, particularly the separation of Electronic Encoding and Data Organization/Coding as already discussed. When these activities are separated there is, for example, a keying operator, and a data organizer who collects and codes the input. If possible, vertical distribution might be extended to the source to attempt to obtain some correspondence between source documents and input requirements. If vertical distribution is not employed, a single operator is given the job of collecting, organizing, and inputting the data.

In TOS the vertical distribution of work is not defined by the operating procedures. The use of preformatted messages implies a data organizer and a keying operator, but in practice, one person frequently does it all. As in horizontal distribution, the vertical distribution of tasks at the ADP interface may depend on operator and environmental factors; i.e. available personnel and work space. In subsequent discussions we will use the term operator to refer to a person responsible for only electronic encoding, e.g. a keying operator; the term user to refer to one who only collects and/or organizes data; and user/operator for one who does both.

The temporal distribution of work refers to the amount of time which the operator spends at the input activity. We will use the term dedicated operator to refer to the person who uses the system full time and casual operator to refer to the person who only interfaces with the system occasionally. The relevant issue is similar to the one involving OJT. The more a user/operator works with the system, the fewer the constraints which are placed on the system software. When the system is only used infrequently, the software must do more to help the user, cover more contingencies, be easier to use, etc.

IV. APPROACHES TO THE DESIGN OF THE INPUT INTERFACE

In designing the man/machine interface for tactical military ADP systems, one must be realistic with regard to the constraints imposed by the ideoyncracies of the military environment as discussed in the previous chapter. In spite of the limitations imposed by these constraints, an enormous range of alternative design features exist which have the potential for matching characteristics of man/machine dialogues with the requirements and abilities of operator personnel.

While all of the internal factors of hardware, software, personnel, and procedures affect the ease of use of the system and, therefore, the production of errors and timeliness of information processed; it is the area of software more than any other which determines how well the interface functions. Given the procedures, given the environment in which the activity must be accomplished, and given the purpose of the system design, it is the software which establishes the dialogue by which the user/operator communicates with the system using the available input hardware. And, for all practical purposes, it is the dialogue and the input terminal which taken together constitute the system, from the point of view of the user. Consistent with this reasoning, the following discussion focuses on characteristics of input dialogue as a method of improving ease of use and reducing the occurrence of input errors.

The discussion which follows purposely divorces itself from the issue of hardware requirements. While some of the dialogue characteristics discussed require minimum hardware capabilities, all can be satisfied by some form of general purpose device programmed to satisfy a wide range of system requirements. This flexibility is essential due to the diverse applications in which tactical military ADP systems are used. In this regard, most of the discussion is written from the perspective of a CRT I/O terminal. This is appropriate for two reasons. First, any input device to be applicable over a wide range of systems must be adaptable to on-line modes of operation, i.e., batch process I/O devices which may have a justifiable purpose in a specific application are not an appropriate device for general purposes. While some mention is made of graphical inputs, action keys, and panels, the system which we are considering will always have a need for variable alphanumeric units. This requirement narrows the choice of an input device to either the typewriter or CRT, of which the latter offers greater speed, reliability, and flexibility for dialogue and error correction.

Although the choice of dialogue characteristics vary considerably with respect to cost (e.g. their impact on system response time, the need for a smart terminal,¹ etc.), our discussion in this chapter focuses only on their impact with respect to ease of use and the occurrence of input errors. It is for this reason that there is no attempt to strongly recommend any specific type of dialogue. The characteristics of dialogue have, in fact, been divided into very basic components with the hope that this will aid the systems analyst in selecting and/or designing the dialogue most appropriate to any given application. The "analyst" can, in effect, "mix and match" any of the techniques or processes discussed to minimize errors within the constraints that are imposed on him. The specific components of dialogues to be discussed include:

- Language
- Formatting
- Automated Processes
- Interactive Processes
- Special Data Entry Techniques

A. INPUT LANGUAGE

The language of a given computer dialogue may be characterized according to its degree of abstractness. Although the underlying dimension of abstractness is continuous, for practical purposes, most examples of dialogue language may easily be classified into one of three basic types:

- English
- Abbreviations/Mnemonics
- Nonsense codes

The degree of abstractness in the dialogue language is equally applicable to labels and to data. Labels, while usually not stored with the contents of data elements, are, when used, an integral part of the interactive dialogue and their level of abstractness may have an effect on the occurrence of input errors.

¹A "smart terminal" refers to an interactive input device which contains a processor and memory.

The choice between the three levels of language depends primarily on consideration of space requirements, time requirements, and the requirements of error prevention. The evaluation of language for any data element must therefore, in a circular fashion, consider the format to be used, method of data entry, length of list of valid codes, and the existence of established abbreviations/mnemonics for the meanings to be coded.

Space is most critical in columnar formats where long labels or data entries may conflict with the required number of columns. Formats which input a single data element have few, if any, problems with space.

The less abstract the language the more characters which are typically needed; therefore, when data entry is by alphanumeric keying, the time required and the likelihood of typographic errors increase as abstractness decreases. When data entry is binary via cursor, light pen or action key, the length of input word has no effect on speed of keying or typographic error. The longer words being less abstract will enhance recognition¹ and thereby decrease glossary errors. This method of data entry is unaffected by recall. Recall failures contribute to variations in input language which in turn contribute to input errors, but only when alphanumeric entry methods are used. It is language variation with alphanumeric data entry which is responsible for many errors in the data organization/coding activity (i.e. glossary, abbreviation, and variable codes) and it is the alphanumeric entry process which is responsible for most electronic encoding errors. However, since this method of data entry has clear advantages with respect to speed and space, and offers the flexibility most often required, the following discussion of language levels is written from the perspective of alphanumeric entry.

English Language

The term English Language, as used here, should be distinguished from the phrase "natural language" which normally implies a nonformatted interactive dialogue with few, if any, restrictions on syntax. An English language dialogue includes any nonabbreviated or coded word, i.e. a data element completely spelled out. It defines the level of language and not the form

¹Recognition and recall are dimensions which characterize how well users can converse in the language. Do they recognize the correct meaning when they see the word and can they recall it when necessary to communicate with the system.

of syntax. English language dialogues may be used with natural syntax or fully formatted inputs. The major disadvantages of English language dialogues are the space requirements and the time required for data entry. The latter, while minimal for a single element may cumulatively become significant. The major advantage of English language dialogue is in recognition and recall. For example, it is far easier to recall the word RETREAT than the abbreviation RETRET, or is it RETREA. While the user may try to substitute the word WITHDRAW he will still learn more quickly with these English words than with their more abstract abbreviations. Where space allows an English language dialogue, communication requirements may be reduced by transmitting only the first 3 or 4 characters of the English word -- whatever is required for uniqueness.

Abbreviations/Mnemonics

This classification of input language includes any abridged version of English words which retain inherent meaning. The major advantage of abbreviations and mnemonics is that they require less space than English language dialogues (although usually more space than nonsense codes) and may be input much faster. Their major disadvantage is that they are generally less easily recalled and recognized than English words although recall and recognition are superior to nonsense codes. With respect to these evaluation criteria they represent a reasonable compromise for input language in most situations. Since there is often little difference in the recognition of English words and their abbreviations, the use of the latter as system-provided labels and field identifiers seems broadly warranted.

The recall of abbreviations and mnemonics can be facilitated by standardizing their length and introducing coding conventions which are consistently employed. Standardizing the length of abbreviations facilitates recall in that the number of coding possibilities is sharply reduced. Extending this same principal suggests that codes with fewer characters will produce fewer alphabetic errors since fewer potential coding alternatives exist. With fewer characters, space is saved which adds to the usefulness of this category of input language. However, while one and two character codes are satisfactory for some data elements such as branch of service, echelon, country, etc., other data elements with extended lists of valid entries require longer codes to provide both uniqueness and inherent meaning.

For example, a two character field, using only alpha characters, allows 676 unique codes, however far fewer codes are

available if inherent meaning is to be preserved. Therefore, when the list of valid entries lengthens, either the length of the mnemonic field must increase or nonsense codes are inevitable. With extended lists of valid codes, the likelihood of error increases. However, as with the English language input, abbreviations and mnemonics are more likely to result in invalid abbreviation errors than the less favorable glossary errors.

The usefulness of abbreviations and mnemonics may be further maximized by implementing the principle of consistency in coding conventions. Consistency is exhibited by TOS, for example, when the same codes are used for branch-of-service in every format where this field appears. However, inconsistency is shown when Armored Tank is abbreviated ARTK as an EN TYPE code and VTANK as a SUBJ code. If TK is to be the abbreviation of Tank, then TK should be used whenever an abbreviation for tank is needed, either by itself or as a component of another abbreviation. If this can't be accomplished, better results might be obtained with nonsense codes.

Nonsense Codes

A nonsense code is defined as an input language with no inherent meaning. Typically numerical entries are used as nonsense codes, although alpha characters can and have been used; typically where more than ten alternatives are expressed in a single character. The principal advantages of nonsense codes are space and speed of data entry. Since inherent meaning is not required, the number of characters in the data element may be limited to the minimum number required to provide a unique code for every valid entry. The disadvantages of nonsense codes are, of course, poor recognition and recall. Given that the list of valid entries is brief and that nonsense codes are assigned consistently between data elements, they can be learned to the point where they are almost as recognizable as abbreviations. In encoding sex, for example, MALE = 1 and FEMALE = 2 probably generates a high degree of reliability through recognition.

Although abbreviations and mnemonics appear to represent the best compromise between space and errors for input dialogue, there are occasions where one of the other language forms should be used. When the list of valid entries is not extensive (e.g. 5 or less) and used frequently, nonsense codes might be as effective as abbreviations and particularly advantageous when speed of data entry is very critical. More over, when used with menu selection formats or glossary displays, nonsense codes are advantageous since they require less space and produce a less

cluttered appearance on the screen. Nonsense codes may also be indicated when the rate of abbreviation errors is high due to recall failures. Such failures may occur either because (1) the operator lacks motivation or (2) he has used the system enough to think he knows what to do. In the latter case errors result because he has not used the system enough to adequately remember the abbreviations/mnemonics. However, if operators continue to guess the reduction in abbreviation errors will be at the expense of an increase in glossary errors.

In these situations, it may be appropriate to substitute non-sense codes for abbreviations to force operators to make use of the glossary. When the list of valid entries is unrestricted, English language inputs are frequently necessary and although abbreviations/mnemonics may sometimes be used in open-ended data elements, English language inputs will generally reduce variations in language and in turn the frequency of retrieval errors. One notable exception occurs with information which people normally abbreviate. This is precisely the problem with Unit Identification in TOS. To adopt English language identifications would be strange and cumbersome, however, undesirable variation exists with the abbreviations and as a result many retrieval errors occur. Coding conventions may be defined in order to reduce language variations and therefore error; however, the more effective route to error reduction is to allow the user to inspect codes which have already been entered.

Quantitative Codes

Quantitative entries can be evaluated on the same dimensions and with the same criteria already discussed. However, the selection of an appropriate language for quantitative data is far easier because the outcomes of the various alternatives are more predictable. The following list shows how similar language categories can be applied to numbers.

Arabic	10000
Abbreviation	10 + 3
Nonsense Code	4

In this example, the abbreviation is given in scientific notation and indicates that the number is 10×10^3 which is equal to 10,000. As might be expected, the nonsense code 4, since it represents an ordinal value, has no meaning without a glossary which might define the numbers less than 100 as 1, from 100 to 999 as 2, from 1000 to 9999 as 3, and 10,000 or greater as 4.

The categories of language applied to quantitative data elements would rank similarly as with verbal data elements on the criteria of space, time, recall, etc. The fact that the software can easily abbreviate arabic numbers suggests, however, that errors would be easily avoided by allowing the user to choose the language of input, using abbreviations only when they are contained in source documents. The choice between English or a nonsense language depends directly on whether or not it is important to preserve the original number or whether a classification scale can be substituted. If exact retrievability is not critical, the choice can be made as if this were verbal data; e.g. how much space and operator time is being saved and how many valid codes will have to be recalled? There is, for example, little saving in space or time from classifying an element such as age, although this may be desirable from the perspective of output and other interpretive or analytic requirements.

B. FORMATTING

The formatting of input specifies the rules by which information is organized on an input medium (e.g. punch card, CRT screen, etc.) in accord with the system's input requirements. Given a level of language as discussed above, how should the language elements be arranged, spaced, and delimited? How much freedom in providing information should be given to the operator and how much assistance should the system provide? Formatting is primarily concerned with three types of information; labels or field identifiers, data, and relational operators. For purposes of inputting data the relational operator of equivalence is usually desired and assumed. Although we may wish to query the system for the location of runways greater than a specified length, when inputting information about a specific runway, the system would normally require that we estimate its length and not its minimum or maximum values. To simplify the discussion we will, therefore, disregard relational operators with recognition that they can be accommodated by similar formatting techniques. The formats to be discussed may be classified into four basic types:

- No format
- Formats with implied labeling
- Formats with explicit labeling
- Formats with codes displayed

The inclusion of labels by the system represents one degree of automated assistance which can be provided to the user, while the display of codes by the system represents an even higher level of computer initiated assistance.

No Format

Two variations of formatting are possible when there is no format provided by the system:

- Natural language
- Unformatted input

A natural language input might be:

The Brazos River Bridge is a 2 lane bridge located at 18QUT75103240 and is 100 feet long, 30 feet wide, and operational.

While the natural language input has obvious advantages with respect to control of input errors, (particularly for casual users) it is typically inadequate to meet minimal reporting and query requirements within reasonable costs.

With an unformatted input dialogue either of the following inputs might be acceptable:

NLANE = 2, NAME = Brazos Bridge, STATUS = OP,
LGTH = 100, WDTH = 30, LOCATION = 18QUT75103240

NAME = Brazos Bridge, STATUS = OP,
LOCATION = 18QUT75103240, LGTH = 100, WDTH = 30,
NLANE = 2

It should be noted that these unformatted inputs make use of both English words and abbreviations for labels and all three levels of language for input data. The unformatted input offers a rapid method of entering data, but without dedicated users is prone to all types of errors. The relational operator is essential in this format as a delimiter between label and data.

Formats with Implied Labeling

Implicit labeling is primarily useful for card or key-to-tape input when explicit labels cannot be provided. Without explicit labels errors of omission and location errors are more likely to occur. Explicit labeling, however, can only be accomplished with

a limited set of input devices, e.g. with a CRT or optical scanning equipment; the latter, not suitable to on-line operations, is not considered in this report. Two formatting methods are used with implicit labels.

- Positional with delimiter
- Fixed format without labels

The information from our previous example input in a positional format with a delimiter might look like one of the following examples:

BRAZOS_BRIDGE,18QUT75103240,OP,,100,30,2

BRAZOS_BRIDGE/18QUT75103240/OP//100/30/2

BRAZOSBRIDGE_18QUT75103240_OP__100_30_2

The double delimiters signify that a variable implied by the input list has been omitted. The choice of a character to use as a delimiter should consider keyboard layout and impact characteristics. Characters should be avoided whose key is difficult to reach (time consuming), too easily reached (provoking typographic errors) or which is used as an input character (provoking location errors). Note that the use of blanks as a delimiter may induce errors as a blank is likely to be used accidentally. Note also that the space between Brazos and Bridge must be deleted when blanks are used as the delimiter. The principle advantage of this input format is in time saved, however, the format is only useful with dedicated operators who will learn the identification and order of data elements. With casual operators this type of format could produce a large number of location errors. Since a similar format with explicit labels can easily be produced with a CRT input device, this format should find little use outside batch processed input media. One frequent application of this format is for dates. Used in conjunction with an unformatted input, the operator might key the following:

DATE = 6/20/30

The user labels the unformatted entry as DATE, while the labels for the subfields of month, day, and year are implied using a positional format without labels. The other style of format with implied labels also has little value for screen inputs. Fixed formats without labels are useful for card inputs where labels cannot be easily provided and the fixed format is desirable to minimize location errors and errors of omission. However, if a

CRT is used for input, there are compelling arguments for the use of similar formats which employ explicit labels. However, to find an example of a fixed format without labels implemented with a CRT display we need look no further than TOS; e.g. the subfields of the SECURITY data field. The format appears as follows:

SCTY: __ / __ / __ ;

The system explicity provides a collective label, but only implicitly provides the labels for the three subfields, i.e., classification, downgrading, and exemption. The fact that each subfield has two fixed characters further removes any potential cue of the subfields identity and increases the likelihood of location errors. The same design characteristic appears several times in TOS. Consider the SUBOR-TYPE field in the UTOA format reproduced in the section on forms completion which follows. This format has excess space in which the subfields of SUBOR-TYPE might be labeled.

Formats with Explicit Labeling

Formats in which the system explicitly provides labels may be categorized into two basic styles:

- Displayed formats
- Form completion

Displayed formats may use fixed or variable length fields. The variable length displayed format is similar to the positional format with delimiter which has already been described. Using that earlier example the CRT screen would display:

ENTER

BRIDGE NAME / LOCATION / STATUS / TYPE /
LENGTH / WIDTH / NUMBER OF LANES

and the user would key in information as shown in the earlier example, i.e.

BRAZOS_BRIDGE/18QUT75103240/OP//100/30/2

Using a fixed length displayed format the screen would display:

ENTER

NNNNNNNNNNNNNN/CCCCCCCCCCCC/SS/T/LLLL/WWW/X/

WHERE

N = BRIDGE NAME

C = LOCATION

S = STATUS

T = TYPE

L = LENGTH IN FEET

W = WIDTH IN FEET

X = NUMBER OF LANES

The displayed format represents an excellent compromise in many situations. It provides a rapid means of data entry with minimal space requirements for the amount of control provided over errors. Displayed formats may be used to enter either a single data element group or multiple data element groups. Since the entry of a single element group may continue on successive lines, there is no serious space constraint as encountered with columnar formats. Omissions are less likely than with either unformatted screens or formats with implied labels. Location errors are less likely to occur with displayed formatting than with implied labelling, however, such errors are more likely to occur with the displayed format than with formats yet to be discussed. Of the two types of displayed formats, the fixed length is preferable with less dedicated users since length of field serves as a cue to the correct code and therefore may reduce abbreviation and glossary errors. To reduce errors of omission, mandatory data elements could be coded using capital letters in the displayed format with lower case letters signifying nonessential elements.

1 : form completion style of format may be established for multiple data element groups or a single data element group. When multiple data element groups are expected, the speed of input can be improved by providing columnar formats. Field identifiers appear as column headings and the user inserts data rowwise for each event. This format using explicit labels, is a simple extension of fixed formats with implied labels adapted to take advantage of the CRT screen. The following example illustrates this format from a commercial accounting system.

AABINT							
TYPE	REF NO	OBJ CODE	ACCT NO	DESCRIPTION	AMOUNT	ADJ	
E	24	301	102465...	CENTRE HDW	150.00		
E	30	371	102544...	GNL STORS	75.00		
E	37	020	102647...	CR 47	1500.00		
A	15	322	102489...		100.00	I	
A	17	336	102490...3		50.00	D	

When the screen is not sufficiently wide to accomodate all of the required fields and a displayed format is contraindicated, the screen can be split with columns in the top half labeled differently from those at the bottom. Alternatively the format can be arranged with the fields as rows and the data element groups entered columnwise. A poor approach to insufficient screen width is illustrated in the following TOS format. This format would appear to increase the likelihood of quantitative errors in the location field.

```
JB+5+;UTDA;ORIG/NO:+++/-----;SCTY:+/ / ;PREC:R;RESTR: ;-----  
-----UNIT-OR-TF-----DEPLOY---TYPE-DATA NAME-OR-NO-----TIME-----  
1: ;-----: ;-----:+++ ;:- ;----- ;:- ;  
2: ;-----: ;-----: ;:- ;----- ;:- ;  
3: ;-----: ;-----: ;:- ;----- ;:- ;  
4: ;-----: ;-----: ;:- ;----- ;:- ;  
1.LOC:+++ ;2.LOC:  
;3.LOC:  
4.LOC: ;-EOT--;
```

If the screen is wider than needed for the number of data elements to be entered and a larger number of data element groups is frequently expected, the screen may be split vertically as in this TOS format

```
J04454;EDXA;ORIG/NO:++++/-----;SCTY:++/ / ;PREC:R;RESTR: ;-----  
MSG-ORIG/NO-----ECH/FUNC--USER--MSG-ORIG/NO-----ECH/FUNC--USER--  
01: / ;-- / ;-- ;--02: / ;-- / ;-- ;--  
03: / ;-- / ;-- ;--04: / ;-- / ;-- ;--  
05: / ;-- / ;-- ;--06: / ;-- / ;-- ;--  
07: / ;-- / ;-- ;--08: / ;-- / ;-- ;--  
09: / ;-- / ;-- ;--10: / ;-- / ;-- ;--  
11: / ;-- / ;-- ;--12: / ;-- / ;-- ;--  
13: / ;-- / ;-- ;--14: / ;-- / ;-- ;--  
15: / ;-- / ;-- ;--16: / ;-- / ;-- ;--  
17: / ;-- / ;-- ;--18: / ;-- / ;-- ;--  
19: / ;-- / ;-- ;--20: / ;-- / ;-- ;--  
21: / ;-- / ;-- ;--22: / ;-- / ;-- ;--EOT--  
;
```

When the number of data elements is excessive for columnar formats, or there is no need to enter more than a single data element group at a time, other styles of form filling formats may be used. Principles of consistency and clarity should be followed in designing an aesthetic and an uncluttered form. Contrast the following TOS formats.

```
J04454;EWFA;ORIG/NO:++++/-----;SCTY:+/ / ;PREC:R;-----  
-----TIME: ;-----  
SUBJ-HEADING:+ / ;-----  
SUBJ-TITLE: / ;-----  
CONCL:+ ;-----
```

;--EOT--

```
J04454;UTOA;ORIG/NO:++++/-----;SCTY:++/ / ;PREC:R;RESTR: ;-----  
UNIT-OR-TF:+++ ;-----SWBD-DSGTR:+ ;--  
-----NATION:US;AR-SVC:A;-----  
-----SUBOR-TO: ;-----SUBOR-TYPE:+++ ;-----  
PARENT:+ ;-----  
-----  
-----  
-----  
EFF-TIME: ;DISTR: ;-----  
REMARKS: ;-----  
;--EOT--
```

Each of the data fields in the UTOA format seems scattered around the screen which makes them difficult to find and difficult to enter data. By comparison, the EWFA format with three of its four data elements on the left margin, appears remarkably superior. With non-tabular form completion formats, consideration should be given to aligning spaces for entering data at the left margin and providing labels to their right or underneath. This makes it easier to position the cursor, without computer controlled tabbing, speeds the entry of data, and, depending on the hardware, limits the amount of information which needs to be transmitted to the CPU. Given a hierarchical structure of data elements, indentation should be considered to reflect the hierarchical relationships.

Other issues to be considered in designing form completion formats are (1) methods for indicating the length of fields and (2) the identification of fields for which the insertion of data is mandatory. Unlike displayed formats, form completion cannot be used with variable length fields without wasting space on the screen (whether the length of the field is shown to the user, however, is optional). One method for indicating length is to insert dashes in the spaces to be filled. In TOS, the beginning of the field is delimited with a colon and the end of the field with a semicolon. Recent technology provides control over the gray scale of the CRT roster which allows a pleasant uncluttered demarcation of the spaces to be filled. With dedicated users familiar with the codes to be entered, it may not be necessary to indicate the length of each field. The cursor may be manually tabbed or automatically tabbed to the first position of the next field as soon as one field is completed. Given the external personnel constraints on tactical military systems, it is probably best to always indicate length of field. Length of field and required data are two information elements which can be transmitted by a single character. TOS fills each space of required data fields with an "M". Given the hardware capability, a reduction in the omission of data elements might be obtained if these characters could be made to pulsate until turned off by the insertion of data.

Formats with Codes Displayed

Formats of the type discussed in this section increase the ease of inputting data by making further use of the flexibility of interactive input devices. With few exceptions, the variations of this format are applicable only to verbal data elements with restricted lists of codes. In general, these formats reduce both abbreviation and location errors, however, care must be exercised to insure that glossary errors are not promoted. The

major disadvantages of these formats are that they consume considerable space, slow down the inputting of information and are extremely demanding on software, memory, and communications. The space problem can be solved by using this format as part of an interactive dialogue where only one element is input at a time, however, the interactive dialogue, discussed below, only makes the input process slower. Two types of these formats will be considered:

- displayed codes
- menu selection

With the displayed code format each data element is accompanied by a list of permissible codes. For example:

BRANCH OF SERVICE _ (A,F,N,M,C)

This format is primarily useful when the list of restricted codes is small and the recognition of abbreviations/mnemonics is better than their recall. For this reason, this format option has no value for use with either nonsense codes or quantitative data. In the example given almost everyone would recognize that F stood for Air Force although they might have difficulty recalling the code. Abbreviation and location errors are virtually eliminated except to the extent that typographic errors occur. The likelihood of typographic errors is low, however, since the format is generally only used with short lists and the abbreviations generally consist of only one or two characters. A variation of this format is found in TOS where question marks appearing after labels indicate that a Y (yes) or N (no) response is required. For example, in the following format:

LIST? _ :

The question mark indicates that the operator should respond with a Y or N to indicate whether or not he wants a list of units. Another variation of this format is to have the user enter a code by positioning the cursor under the code desired and pressing a transmit key. The system should then enter the abbreviation in the space provided for verification. This technique eliminates abbreviation errors with a moderate risk of increasing glossary errors (cursor positioned incorrectly and operator fails to verify code selected). Speed of inputting is slowed, however, since the user must position the cursor.

If the list of valid entries is still relatively short, but recognition of codes is poor, then the menu selection type of

format is appropriate. For example, the screen might display the following:

ENTER PRECEDENCE

F = FLASH

I = IMMEDIATE

P = PRIORITY

This format is also used with three methods of data entry. The operator may have to key the correct code into a space provided, position the cursor under the correct code or key the correct code anywhere on the screen. While cursor positioning may eliminate some errors caused by typographic mistakes (i.e. abbreviation and location errors are impossible), keying the correct code anywhere on the screen is a faster method of entry.

This format would seldom be used for insertion of an entire data element group or message set from a single display, yet, given the proper types of data elements it could be used to input an entire message set one element at a time as in an interactive question and answer dialogue. Since the format cannot be used for quantitative data (except with numbers categorized by ordinal scale) it is more likely, in military applications, to be used for individual data elements as opposed to entire message sets. While it may be difficult to display an entire data element group or message set in this format in one screen, a single screen may be used to format more than one data element or to permit more than one response to the data elements presented. An example of a screen used to input two data elements is suggested by the following example using accuracy and reliability codes.

ENTER RELIABILITY/ACCURACY

RELIABILITY CODE	ACCURACY CODE
A Completely Reliable	1 Confirmed by Other Sources
B Usually Reliable	2 Probably True
C Fairly Reliable	3 Possibly True
D Not Usually Reliable	4 Doubtful
E Unreliable	5 Improbable
F Reliability Cannot Be Judged	6 Truth Cannot Be Judged

C. AUTOMATED PROCESSES

Given the enormous capability which the computer offers, it is important to consider what specific processes might be implemented through software to control the occurrence of input errors. While these processes are certain to add to overall processing time unless "smart terminals" are introduced, their payoff in error reduction may be considerable and worth the cost. The three processes to be considered include:

- Error Detection
- Editing
- Data Base Update

Error Detection

Every system has some minimal amount of validity checking to detect errors and insure that it can maintain control in spite of erroneous instructions from the user. The extension of validity checking to data depends on the nature of the data element. Unrestricted or open-ended data elements such as names of people, places, organizations, and documents are not amenable to validity checking since a list of restricted codes cannot be established and the user is free to input any new name. Also, validity checking is difficult though not impossible for quantitative elements or for elements with extremely long lists of restricted codes. Validity checking, however, can be applied to any data element whose input can and should in some way be restricted. Verbal data elements should, when possible, have a list of restricted codes for all data elements which may be queried; there is, on the other hand, no need for restricted codes or validity checking in fields which will not be queried (e.g., remarks) and, of course, restricted codes cannot be established for open-ended data elements.

Validity checking can be implemented with any level of language, although English words should be avoided unless very short. Depending on their uniqueness, it may be possible to allow the user to input long English words and for the system to store and check only the first four or five characters. Typographic or recall mistakes will normally produce detectable errors, except with consecutive nonsense codes when mistakes will often result in a valid code. When the error is caused by a careless mistake in checking a glossary, an undetectable glossary error is likely to result, whatever the language of the data element.

The longer the list of restricted codes, the more recall problems the operator will have and errors (abbreviation and/or glossary) are more likely to occur. One suggestion for reducing abbreviation errors is to expand the list of legal codes for each meaning, so that codes previously categorized as invalid, but which communicated the correct information (i.e. have high recognition value) are acceptable to the validation process. For example, the legal codes for TANK could be expanded to include TK and TNK. Conversion to a common code for storage would be made by the computer. The same process can be extended to the misspelling of English words e.g. TINK for TANK. This solution, which has appeal to natural language advocates, suffers from many of the same difficulties. Alternative codes accepted would have to be unequivocal in meaning (e.g. TK is not an acceptable abbreviation for TANK since it could also refer to truck. Equivocations, however, are likely to be greatest in data elements with long lists of coded meanings, i.e. those elements with the most serious recall problems. Also, when the data element has a long list of codable entities or meanings, the list of valid entries with more than one acceptable code per meaning becomes lengthy and therefore costly to process. Glossary errors might very well increase since codes input would have a higher probability of being a valid code for a different meaning. This is particularly likely to happen when the error is the result of a transcription or typographic mistake. The procedure is also dangerous in that it allows the system to accept mistakes, it may well result in their encouragement; not only in the data element in which they are allowed, but other data elements as well. Therefore, while increasing the list of acceptable codes may reduce the occurrence of abbreviation errors, it has the potential of circumventing the purposes for which validation is implemented.

Assuming that interactive processes are not available to assist the user, several other alternatives exist which could be implemented separately or together. The first alternative would be to standardize the rules for generating codes so as to restrict the number of possible variations the user might generate. Such standardization includes length of field, use of abbreviations within mnemonics, and rules of order, first letter, etc. For example, TOS uses RR consistently in different mnemonics to signify railroad; yet BD, BDR, and BDRY are all used to signify boundary. The next alternative would be to encourage casual users to make use of a glossary, and provide accessible copies. Finally, the system can be programmed to increase the cost of validation errors to the user and thereby raise the level of certainty which he requires before he trusts his recall. The cost of validation errors can be increased by erasing a part of the

input before returning it to the user for correction. The principle behind this is recognized by programmers whose reliance on software to assist in program "debugging" increases as turnaround time decreases and peaks with an interactive compiler. While this procedure may appear self-defeating, it should be recognized that it is only recommended for situations where system overload, caused by repetitive entry of inputs from validation errors, is already adversely affecting the system performance. While the bright and motivated user might recognize that he can input data quicker by manually checking codes than using system validation to find his mistakes, the less motivated user will respond sooner with some behavior modification. The system may also experience a reduction in glossary errors by encouraging users to manually check codes before the code checking becomes warranted purely on the basis of turnaround time.

If a mathematical algorithm can be formulated to check the validity of quantitative data, the number of quantitative errors may be reduced. Time is an example of a quantitative element which can be checked for valid codes in the range of 0001 to 2400. Unfortunately, with many quantitative elements, validity cannot be expressed beyond the constraints imposed by the number of characters provided. Therefore when 3 characters are specified, all numbers up to 999 are usually valid. Probably more can be done to reduce quantitative errors with a process we will call conditional.

Conditional validity checking is the process whereby the system changes the list of restricted codes depending on the value of other data elements, e.g. if A = Z then the restricted list for B includes only Y and Z. If the operator inputs A = X and B = W the system can indicate that an error exists although it may not be able to determine which data element is wrong. This process which may significantly increase system costs, has the potential of detecting a large number of errors not otherwise detectable. For example, with conditional validity checking the system could invoke different valid codes for equipment lists depending upon which Branch of Service or type of unit is input. Conditional validity checking also enables many more quantitative errors to be detected. For example, with unconditional validity checking the system may have to accept any 4 digit number greater than 0500 for runway length. However, once the user inputs a type of aircraft that requires more than 500 feet, the system equipped with conditional validity checking can reject runway lengths less than the actual minimum requirement of the designated aircraft. The information for conditional validity checking may come either from the same input as the data being

checked or from data previously entered, given a summary data base. For example, by assuming maximum rates at which units of different types and sizes can move, the validity of time and location data can be checked given an earlier time and location in the data base.

If the conditions which restrict the lists of valid codes for some elements are likely to remain fixed over a period of time, the list of restricted codes can be changed by the user as appropriate and error detection could proceed without the need for a conditional validity checking capability. A similar approach could be taken when the conditions are fixed for different users/terminals particularly if smart terminals were available. The user could input via tape or floppy disk the lists of restricted codes for the conditions which would remain fixed for a period of time. When those conditions change or the operator "signs off" the terminal, a new set of restricted codes would be read in.

Two additional types of error checking should be mentioned. Adaptive error checking introduced by Gilb and Weinberg recognizes that the rules for detecting errors may change with other events.¹ The system may be programmed to make the necessary adjustments when appropriate or to have the user manually change the rules as his experience dictates.

Another form of error checking is probabilistic. The system can easily be programmed to report the a priori probabilities based upon expected frequencies when known. In this way the system can respond with a probability of the incorrectness of an otherwise valid entry. The technique has its greatest utility in the elimination of all types of quantitative error. This same process can be extended to use conditional probabilities in the same way that validity checking can be made conditional. Both processes can be made adaptive by having the system report empirical or exact probabilities instead of a priori probabilities. For example, the a priori probability of ammunition use by a FA Battery of X or more rounds per hour might be .001. Therefore, given an input of X, the system might respond with the following message.

RATE OF USE UNLIKELY

ONLY 1 BTRY IN 1000 REPORT THAT MANY RDS/HR

¹ Gilb, T., & Weinberg, G. M. Humanized Input Techniques for Reliable Keyed Input. Winthrop Publishers, Inc. 1977.

Given an adaptive process the system could compare observed and theoretical distributions and report significant discrepancies to the system controller. The system controller would then have to investigate and decide whether the the data base is erroneous or whether the a priori probabilities should be changed.

Editing

The process of editing is directly linked to formatting. Editing processes allow the user to input data in a format easiest for him and to have the data converted by the system into the format required for storage. Editing is in effect an edit prevention process, while validity checking is error correction. Some basic editing processes already mentioned include the elimination of labels before storage of data, and the stripping of all but the left-most characters of data input. A large number of the other editing processes can be programmed into software and should be considered. For example, editing can be used to space subfields of data stored as single elements so that the user is more likely to see his error. The process is particularly applicable to data elements such as dates and coordinates. Fewer errors are likely when the user inputs 9/27/79 than when he must key 092779. Editing gives the format designer considerable freedom in spacing fields to achieve consistency formats and an uncluttered appearance within.

Another editing process which reduces errors is the right justification of numbers. When an operator fails to fill all positions with numbers he is much more likely to omit leading than trailing zeroes and the software can easily make the required adjustment.

Still another editing process which is aimed at the reduction of quantitative errors is metric conversion. When it is possible that the sources of information will provide numerical data in more than one scale, consideration should be given to allowing the user to input both the number and scale and have the software accomplish the conversion. Other mensuration processes might be applied to obtain calculated or derived measures; e.g. distances, areas, volumes, percentages, etc.

Data Base Modification

When a summary data base is maintained and detailed records have no value other than for purposes of tracing errors and system monitoring, it may be advantageous to have the operator input data directly into the data base. In practice, he inputs data into a

screen image of the data base and the data base is not updated until edit and validation are complete. A record of the changes can be maintained separately for historical purposes or file recovery.

Data base modifications can be accomplished with any level of language, almost any format and with or without interactive processes. Working directly with a data base image provides a concise method of accomplishing additions, changes, and deletions in a single format. Providing the user with a view of the data base contents at the time of input is at once a method of reducing errors of omission and errors of commission.

When making additions, changes, or deletions to the data base, the user can examine the contents of the data base to verify that he is adding information into the correct record. With data base modification the user can also inspect the current information for accuracy and note when the information is wrong, e.g. a recent change in a units location has not been made. The process is applicable to all types of data bases, but finds its greatest value with hierarchical or vertical file structures.

D. INTERACTIVE PROCESSES

It is the purpose of this section to discuss the role of various interactive processes in the control of input errors. Given an interactive terminal there are a number of procedures which can be established to assist the operator inputting data. Although level of language and formatting were discussed and evaluated in the context of an on-line CRT terminal with interactive capabilities, none of the techniques discussed under language and formats require an interactive interface for their implementation. In fact, most of the procedures already elaborated upon could even be implemented using one or more batch-processed input methods.

Before beginning a discussion of some basic interactive processes, it is appropriate to define the minimum requirements for an interactive capability. All systems are in one sense interactive, but not all are said to have interactive capabilities. Even those with on-line I/O terminals may or may not be interactive. Systems not considered interactive have relatively long response times often referred to as turnaround or delivery time. On the other hand, systems typically referred to as interactive are characterized by extremely brief I/O transaction times. These transaction times have two components: (1) the system response time, and (2) the user response time. A system may fail to be interactive because one or the other of these components

is unsatisfactorily long. The system response time may be too long because of poor design, inefficient coding, or too many users with too many demands. The user response time may be too long because the user is uncertain of his response or is required to do too much.

Two authors (Nickerson 1969 and Doherty 1979) have commented on the requirements for system response time in interactive systems; Doherty accepting a maximum of 2 seconds with subsecond response time being the desired norm.¹ Nickerson calls attention to the fact that variance in response time may be as important as the mean.² Therefore, to reduce uncertainty about when the system will respond, unusually short responses should be avoided. The system, by introducing a delay, can reduce the variance of system response time and uncertainty that goes with it. The broad disagreement in what constitutes an interactive system is undoubtedly related to application of specific characteristics including the type of interaction, operation, and work procedures. For example, a user will frequently tolerate a longer system response time when waiting for an answer to a complex query than when waiting for confirmation that an input was incorrectly received.

Equally important to the achievement of an interactive dialogue, in our opinion, is the length of user response time. Three things which the system can do to minimize this component are: reduce system response time which effects user response time through loss of user attention, help the user when he becomes uncertain of his response, and respond to user uncertainty or error quickly. How these requirements are satisfied depends, in part, on the hardware. For example, in half-duplex systems the user must complete an input before the system can respond; while in full-duplex systems, the computer can respond to each character keyed or every discrete position of the cursor, etc. With a half-duplex system a conversational dialogue is essential if the system is to be interactive, however, with a full-duplex system, the system designer has far more latitude. The trend to smart terminals, given the growth of microprocessor technology and the proliferation of minicomputers, should make it easier to provide an interactive interface since the demands on communication

¹Doherty, W. J. & Fischer, C. "Human Factors: Impact on Interactive Computing". A seminar presented January 16, 1979.

²Nickerson, R. S. "Man-Computer Interaction: A Challenge for Human Factors Research" Ergonomics, Vol 12, No. 4, July 1969.

channels and the main CPU can be dramatically reduced by shifting some of the processing load to the terminal. The following interactive processes which hold promise for the reduction of input errors will be discussed:

- Conditional formatting
- Glossary display/"HELP"
- Expanded definitions
- Conventional Dialogue

Conditional Formatting

The processes discussed in this section represent extension of the formatting techniques already described. Conditional formatting procedures may be applied to either formats with explicit labeling or formats with codes displayed.

Conditional formatting applied to formats with explicit labels can be used to change the information provided to the operator concerning required data elements, and default values. With respect to the required data elements the screen might display:

ENTER

INSTALLATION NAME/LOCATION/STATUS/TYPE

The operator keys in:

BRAZOS_BRIDGE//OP/1

The system may then respond:

ENTER

LENGTH/WIDTH/NUMBER OF LANES

Other methods could be employed with form completion formats for the same purpose: characters indicating mandatory entries could be inserted; labels could be rewritten as capitals, or, made to pulsate, etc., as their preconditions are met.

Although default values can be provided by formats without interactive capability, default values which are conditional on information input cannot be formatted without a highly interactive

dialogue. Given interactive capabilities, glossary, and retrieval errors may be reduced by presenting default values as a function of information input in other data elements. For example, input formats established to enter intelligence information from various sources might, if the majority of traffic is from Army sources, have default values as follows:

<u>DATA ELEMENT</u>	<u>DEFAULT</u>
Source	A
Coordinate System	UTM

Should the operator change the source code from A (Army) to F (Air Force) the default value could be promptly changed by an interactive system to LAT-LONG. This process is most easily accomplished with a forms completion format.

Another conditional default format exists, which is suitable to columnar forms completions. When it can be anticipated that one or more columns of information may go unchanged from line to line, it may be helpful to allow the system to use the value from the previous line as a default, i.e. automatic ditto. This process reduces the amount of data to be keyed and therefore the likelihood of all types of errors. For example, on the following screen image, unit name and supply class are pre-programmed for automatic ditto. Entries within the boxes were default values inserted by the system.

<u>UNIT NAME</u>	<u>SUPPLY CLASS</u>	<u>SUPPLY TYPE</u>	<u>UNITS</u>	<u>AUTHORIZED</u>	<u>ON HAND</u>
5_FA_BN/3_FA_RGT	AMMO	30CAL	CASE	125	125
5_FA_BN/3_FA_RGT	AMMO	50CAL	CASE	250	125
5_FA_BN/3_FA_RGT	AMMO	105MM	ROUND	3200	1600
5_FA_BN/3_FA_RGT	POL	DIESEL	GAL	6000	3000
5_FA_BN/3_FA_RGT	POL	10W30	QT	500	400

Glossary Display/"HELP"

When the list of valid codes is lengthy, the menu selection format becomes cumbersome. A more appropriate dialogue given an

interactive terminal is the glossary display or "HELP" routine. The operator working with one of the other format types enters a question mark or other identified in the field for which he wants the computer to display the valid codes. In response, the glossary appears on the lower portion of the display. If necessary to provide sufficient space, the system can temporarily erase all lines from the existing format except the one on which the operator is currently working. If the data is to be keyed, it is essential that both the glossary and the format for the relevant data element be displayed simultaneously so that the operator is not forced to memorize the code. Alternatively, it may be advisable to use the "X" to select (i.e. binary) form of data entry. When the glossary cannot be displayed on a single screen image, the operator can view chunks (successive pages) or scroll the display. Conditional relationships expressed for validity checking may also be used to reduce the amount of the glossary which must be shown. Still another variation would be to have the operator enter the first one or two characters of the code he requires before striking a "?". The system can then branch to and display the relevant part of the glossary. Using this technique, data elements with unrestricted codes can also be accommodated. For example, if the user entered the first two characters of unit name, the system could display a list of all units already entered with names having common first letters. If the user cannot find the unit name he requires, he might try an alternative spelling or enter a new name following established conventions.

Both menu selection and glossary display formats can be used with any level of language. When the data element has a list of restricted entries, mnemonic codes or abbreviations should be favored, particularly when the user must key in the code, and, the menu or glossary displayed is compact. When scanning a display, the operator might inadvertently select the code above or below the one he wants: with nonsense codes, this error type would go undetected and a glossary error would occur; mnemonics, while taking slightly longer to input, and perhaps with a slightly higher likelihood of an abbreviation error, lower the possibility of the more serious glossary errors since the operator has some chance of seeing his mistake.

Expanded Definitions

There are two types of interactive processes which can be characterized as expanded definitions. In one process the system converts abstract inputs into their full English (or arabic) equivalents. In the other process, the system provides the operator with additional information from the data base about the

subject of the input data set. The first process is very useful for reducing the occurrence of glossary errors, while the second process is most useful for limiting all types of errors with unrestricted data elements. For either method to be effective, the system must be interactive at the data element level and the data organization/coding activity should take place at the input terminal. If these conditions are not met the computer is unlikely to use the information to verify his input and the capability will be wasted. If the system is designed so that it cannot respond to a user input until after an entire data set is composed and transmitted, it is unlikely that the operator would take the time to verify each entry since he is otherwise ready to move onto something else. Similarly, if the operator has the input organized and coded on a hard copy format, he is very likely to act as a typist and not question the accuracy of the information he is entering.

Expanded definitions of the first type can be used with any abbreviation mnemonic, nonsense or ordinal code for which a restricted list is established. After the operator keys an input, the system responds with the full definition. This process is most easily applied with positional formats which have variable field lengths that provide adequate space. However, space could be provided in form completion formats for definitions which are relatively short or, alternatively, definitions might be provided at the bottom of a split screen. As an example, should the TOS operator key in "DEFLT" the system would respond "Enemy Front Line Trace." If he should be entering a friendly front line trace, the error would be detected and "DFFLT" could be entered. Another version of the same process is for the user to input a unit or file number and the system responds with the identical unit or file name.

Just as the first type of expanded definition requires an a priori list of restricted codes, the second type requires the maintenance of a summary data base. The procedure is analogous to a data base modification process (discussed earlier) where the user inputs directly into an image of the data base file. In the expanded definition process, information is retrieved from the data base and inserted in appropriate places of the detailed input format. For example, if a summary file is maintained for each tactical unit, once the unit ID is entered into a format, the system may respond with other elements of information, including unit location, authorized equipment, etc. If the system does not have a unit with the same ID, the user is alerted and retrieval errors are avoided. If the user inserts a Unit ID which the system associates with some other unit; information

about its location and authorized assets may make the user aware of the mistake and avoid the serious error of disseminating incorrect information.

Conversational Dialogue

The highest level of interactive processing is achieved with the conversational or question and answer dialogue. The principal need for this form of dialogue is as a means of implementing other forms of interactive processes; in fact, depending on the hardware, the conversational dialogue may be the only acceptable method of implementation. If the system cannot react to an input without the user striking a transmit key, entry of one data element at a time is virtually dictated if conditional formatting is to be provided. Even when the system can react to inputs without a transmit response, the changing of formats while the user is entering data may be distracting. Furthermore, since it is difficult to display a glossary for more than one element at a time, this interactive process dictates a relaxed input pace even when the entire format can be displayed. It is often desirable that the screen be split to build-a-record; i.e. the user is allowed to see at all times his answers to earlier questions.

The conversational dialogue is obviously a slow method of data input, and because of this it must do more in terms of reducing errors and elimination of unnecessary data entries if the user is to tolerate the slower pace. The utility of this form of dialogue will essentially depend on the extent to which conditional formatting is useful and the frequency with which users request the glossary or "HELP" processes. Therefore this type of dialogue is particularly useful with hierarchical data bases where the questions (data elements) at each level depend on data entered at the next highest level and data base modification dialogues are not desirable. When the number of conditional data elements is small and the majority of users can recall the majority of codes without the computers aid, the conversational dialogue is not likely to be justified.

E. SPECIAL DATA ENTRY TECHNIQUES

The techniques discussed so far have focused on the entry of alphanumeric data from the perspective of a flexible input terminal such as a CRT. Other data entry methods including binary and graphical are possible and may in special circumstances be more appropriate. When the potential responses are finite and can be built into the software, the potential for binary (Yes/No) input exists and was, in fact, suggested in the example of data entry

by cursor positioning with a menu selection format. This type of data entry can be made more effective with the addition of a light pen capability which improves the speed of input and reduces errors caused by typographic mistakes. Justification for this form of data entry will of course depend on the number of applications which the light pen has. The flexibility of this alternative can be increased by including numbers and letters on the screen which can also be selected via the light pen, but to do so consumes valuable display space and increases the likelihood of errors. More often it will be advantageous to use the light pen in conjunction with a keyboard.

Binary information may also be input with the application of action keys or panels. Action keys may be employed by providing special labels to keys on a standard keyboard or by building appropriate dialogue into a special purpose piece of hardware. The latter approach is a worthwhile solution in a well defined application which is unlikely to change and for which many copies are desired, a condition which hardly typifies the tactical military system requirements. The labeling of keys on a standard keyboard with special meanings, an approach worth considering, is, however, limited by the number of available keys.¹ Typically, therefore, this technique finds its most useful application in initiation queries, choice of applications, exercise of control functions, etc. The only other application which action keys are likely to have for data input are in the entry of field identifiers with an unformatted input. The following examples indicate how the light pen and action keys might be utilized to input changes in Unit Authorized Assets with TOS. This and the relocation of units are the two most frequent types of inputs made in the operations area.

First the operator would strike an action key that indicates he wants to address the unit status file. The system might display the following:

UNIT IDENT - - - - -

<input type="checkbox"/> PERSONNEL	<input type="checkbox"/> ADD
<input checked="" type="checkbox"/> EQUIPMENT	<input type="checkbox"/> CHANGE
<input type="checkbox"/> CRITICAL SUPPLIES	

¹Overlays can be used to reassign meanings for different applications, but this should not be required on a frequent basis.

The user keys in the Unit ID, indicates the boxes checked with a light pen, and the system displays:

EQUIPMENT CODE - - - - - HELP

EFFECTIVE STRENGTH - - -

AUTHORIZED STRENGTH - - -

If the operator had checked CHANGE instead of ADD, the system could respond with the active numbers in the data base. If the operator had checked CRITICAL SUPPLIES, the system would display:

<input type="checkbox"/> SMALL.ARMS AMMO	% OH - -
<input type="checkbox"/> MORTAR AMMO	CBT LOAD - - - - -
<input type="checkbox"/> ARTILLERY AMMO	
<input type="checkbox"/> TANK AMMO	
<input type="checkbox"/> DIESEL	
<input type="checkbox"/> GASOLINE	
<input type="checkbox"/> JP-4	
<input type="checkbox"/> FOOD	
<input type="checkbox"/> WATER	

thus, conditional formatting is implemented. The fact that Unit Identification, strength, percentage, and load numbers do not have lists of valid codes, however, requires that either the numbers 0 through 9 be displayed, or a keyboard is available. Dialogues using light pen input are by necessity interactive and therefore can be tailored to very casual and untrained users.

With action keys the system would again have the operator strike a key which indicates he wants to address the unit status file. Each of the labels in the above formats would have corresponding action keys labeled with a keyboard overlay. Labels in the same class should be grouped together on the keyboard. After entering header information and Unit ID, the user, by striking

two keys would, for example, have displayed:

PERSONNEL = (EFF-STRGTH = _ _ _

the operator would then key three numbers (e.g. 250) and another action key and see the following display:

PERSONNEL = (EFF-STRGTH = 250, AUTH STRGTH = _ _ _)

He would then type the numbers for authorized strength and another action key for equipment or critical supplies, etc.

A dialogue using action keys may or may not be interactive. Validity checking can be accomplished as each data element is entered or when the entire input is completed. Glossary displays can also be used to aid the user in recalling data element codes. This type of dialogue, however, does require the same level of user capability as the unformatted dialogue with all alphanumeric data entry. One advantage of the action keys is that they provide, for the user, a more rapid and error proof method to input data with a limited number of fields which are used repetitively.

Although binary input shows promise for data entry applications, the potential for graphical data input is far more limited. Martin lists four impediments to the emergence of graphics at an interactive terminal:¹ graphic terminals are (1) designed for elaborate requirements, (2) require extensive software, (3) focus on engineering applications, and (4) require excessive bandwidth for teleprocessing applications.

While technological advances have negated most of these problems (witness the AN/UYQ-19), the primary advantage of these terminals is for output and query applications and not data entry. One potential application given adequate resolution and scale would be to use graphic capabilities with map overlays for inputting or changing unit locations. Even here, however, it would be difficult to accrue any major advantage over alphanumeric entry of coordinates unless a large number of units were to be located.

¹ Martin, J. Design of Man-Computer Dialogues. Prentice-Hall, Inc. N. J., 1973.

F. SUMMARY

It has been the intention of this chapter to identify a number of input techniques which can be used in the design of man-machine dialogues to improve ease of use, speed of input, and the accuracy of the data input. The focus has been on the input activities of data organization/coding and electronic encoding. The assumption has been that with an understanding of the effect of various inputting techniques on human performance, the system designer can tailor the input dialogue to the specific external and internal constraints of a single application. The point of view promoted is that not only will one technique not fit all applications, but that a single technique is not likely to be applicable to inputting all data in any one system. The designer concerned with optimizing the man-machine interface will evaluate each input requirement separately. Different data elements may require different language, different computer processes to provide a cost effective solution to the inputting of data.

While the occurrence of errors is a serious problem often caused by ignoring the limitations of the operator in the design of a dialogue, Hammer correctly warns against an overzealous approach to the elimination of errors.¹ The more elaborate error prevention and detection procedures may have a negative impact on system responsiveness which may in turn negatively impact on user acceptance and the production of errors from boredom or lack of confidence with the system. The elimination of errors must therefore not be seen as an end in itself. Errors have both "sensitivity" (e.g. how much does one error effect an overall report), and "impact" (i.e. what effect does an erroneous output have in terms of wrong actions/decisions). These factors must be considered and the costs of their eliminations weighted against their potential damage.

The potential counterproductivity of eliminating errors (commission implied) has a related problem in the elimination of omissions. In the first case we may try to do too much, in the latter, too little. Failure to filter incoming data and failure to purge data bases can either saturate the user with information or cause unsatisfactory delays attempting to process and assimilate enormous quantities of data.

¹Hammer, M. "Error detection in data base systems"
National Computer Conference, 1976.

V. COST BENEFIT ANALYSIS

A. INTRODUCTION

The error analysis activity results not only in the identification and classification of error types, but also in a determination of error sources; given this determination causal links are established or inferred and remediation alternatives are suggested. In some instances the alternative may be designed to prevent the error event, e.g. an interactive process to assist in the recall of correct mnemonics; in other instances, the solution may be "remedial" in a more direct sense, i.e. error-detection edit checks may be applied via a software solution to eliminate or reduce the effects of a specific input error given that it occurs. It is clear that for any error, a variety of solution alternatives may be available: moreover, several alternatives may exist among and between broad remediation classes (Hardware; Software; Personnel; Procedures) for the same error.

The enumeration of solutions (or remediation alternatives), then, stems from the error analysis activity and leads directly to an equally complex task -- that of choosing the "best fix" possible given the external and internal constraints that are imposed by and on the decision-maker. "Best fix" in this sense can be somewhat loosely equated to that solution which meets the objectives most effectively within the given limitations.

The military establishment, better perhaps than any other entity, appreciates the complexity of this type of decision process. Indeed, DOD Instruction 7041.3, "Economic Analysis of Proposed Department of Defense Investments," requires the application of cost/benefit analyses in military investment decisions.¹ To this end, the Army and its sister services have supported the development of a number of mathematical guides, tools, and techniques designed to overcome the most serious inherent limitation in the cost/benefit decision process, viz., a reliance upon assumptions -- some that are fully or partially quantifiable, others that defy quantification. Because of these necessary assumptions, experience plays a critical role in "value" assignment with the somewhat mixed result that subjective

¹ U.S., General Accounting Office, The Comptroller General, Impartial Cost-Effectiveness Studies Found Essential to Selecting New Weapons; DOD B-163058 Report to the Congress, 21 August 1972, p. 8.

judgment is a major source of both data and (unfortunately) error. The Army's Engineering Design Handbook does an excellent job of presenting the multiplicity of sophisticated mathematical procedures and techniques that are available to deal with identified uncertainties,¹ therefore, its critical review is highly recommended.

There is, however, a more than acceptable alternative to the rigorous and extremely time-consuming approach taken in the Handbook, one that leads to decisions with considerably less effort than the process implied in the Handbook, and one which has the flexibility to accept in its application, certain of the mathematical procedures and techniques which render judgment more meaningful. The approach is "Multi-Attribute Utility Measurement" (MAUM).

It seems fruitful at this point to defer a discussion of the mechanics of the MAUM approach until after a suitable perspective has been provided in more conventional terms. Thus, the balance of this chapter will deal with the development of a context for cost/benefit analyses, the role of the cost/benefit analyst vis-à-vis the decision-maker, a discussion of the major factors to be considered (objectives, costs, and benefits), illustrative methods for dealing with uncertainties and disagreements, and finally the MAUM process.

B. COST BENEFIT ANALYSIS: THE CONTEXT

The selection of a particular alternative (from the solution space being considered) as a "fix" for a specific error can be characterized quantitatively as a "cost-benefit analysis", a "benefit/cost analysis" or a "cost-effectiveness analysis" depending upon the preferences and specific aims of the analyst.

All three are essentially the same insofar as the analytical objective is concerned -- to provide a rational, systematic quantification aid to the decision-maker.

Cost benefit analyses are in the opinion of Prest and Turvey² "... a practical way of assessing the desirability of projects, where it is important to take a long view (in the sense of

¹AMCP 706-191; Engineering Design Handbook: System Analysis and Cost-Effectiveness; U.S. Army Material Command, Headquarters, Washington, D.C., 9 April 1971.

²Prest, A. R. and Turvey, R., "Cost-Benefit Analysis: A Survey". Economic J., Vol. 75, No. 300 (Dec. 1966).

looking at repercussions in the further as well as the 'nearer', future) and a wide view (in the sense of allowing for side effects of many kinds ...); i.e., it implies the enumeration and evaluation of all the relevant costs and benefits." More pointedly, Quade states that "... it is an analytic study designed to assist a decision maker identify a preferred choice from among possible alternatives."¹

In a fairly comprehensive state-of-art review, Pagano and Sauerlander assert that,² "... it is a way to look at a problem, analyze it, and arrive at some type of solution. It involves the comparison of various alternatives to achieve a specific objective and essentially consists of the following six steps:

1. The statement of the desired objectives
2. A complete specification of all the relevant alternatives
3. An estimation of all the costs involved
4. An enumeration of all the benefits
5. Development of a model, either verbally or mathematically
6. Development of criteria for choice among the relevant alternatives."

These authors caution that these "steps are so interrelated that any attempt to discuss them as mutually exclusive parts is doomed to failure".

The consideration of cost-benefit analysis as a method for selecting a remediation alternative to "fix" bad information resulting from input errors is somewhat paradoxical, perhaps even ironic since the more serious criticisms of cost-benefit techniques assert that they deal with imperfect information in

¹Quade, E. S., "Cost Effectiveness: An Introduction and Overview." The RAND Corp., P-3134 (May 1965).

²Pagano, A. M. and Sauerlander, O. H. "Benefit-Cost Analysis" in Roadway Delineation Systems, NCHRP Program Report 130, 1972, pp. 156-186

imperfect ways. Quade, for example, cites¹ "... the necessity that measures of effectiveness be proximate" as a serious disadvantage and goes on to enumerate other shortcomings in applying the technique: (1) "Limitations on time and money obviously place sharp limits on how far an inquiry can be carried"; (2) "We can't be as confident that our estimates of effectiveness are essentially correct as we are about our cost estimates; ... the analysis can never treat all the relevant factors", and (3) "No matter how thorough, it always leaves something for the decision-maker". Levine would restrict the use of analysis to uncovering large quantitative differences only and cautions that decisions should not be based on small differences.² His position is that "For one thing the numbers used in systems analysis are always imperfect and to make decisions on the basis of small quantitative differences derived from very fuzzy inputs is wrong and is dangerous. If differences are small, then an entirely different basis for decision should be arrived at. Indeed, if quantitative results do not accord with one's intuition, one had better check his numbers very carefully, because by and large intuition is the better guide".

Bell in a less loaded fashion, acknowledges the role of intuition and at the same time lends strong support to the utility of Cost/Benefit analysis as an adjunct to the decision process:³ "There is no substitute for experiment, experience, intuition, and judgment, all of which can still lead to wrong answers. The identification, quantification, and systematization of cost-effectiveness analyses can, however, add to the likelihood that the judgment-decision is a good one."

Although cost-benefit approaches fall far short of demonstrating that a particular course of action or a particular alternative remediation is best beyond any reasonable doubt, the technique according to Quade⁴ "... is able to make a more

¹ Quade, E. S., "Some Comments on Cost-Effectiveness." The RAND Corp., P-3091 (Mar. 1965)

² Levine, R. A., "Systems Analysis in the War on Poverty." A paper presented to the Operations Research Society of America (May 1966).

³ Bell, C. F. "Cost-Effectiveness Analysis as a Management Tool." The RAND Corp., P-2988 (Oct. 1964).

⁴ Quade, "Some Comments".

systematic and efficient use of judgment than any of its alternatives." If nothing else, the analysis can eliminate really bad alternatives and provide for the decision-maker a smaller solution space, i.e., a shorter list of potential remediation alternatives, to choose from.

The Comptroller General of the United States, in a report to the Congress¹ on the use of cost effectiveness techniques to select weapons systems supports the technique, but points to the effect of its inherent limitations (assumptions, uncertainties, etc.) and concludes that cost-effectiveness determinations should be considered as an aide to the decision-maker rather than a document that indicates which weapon should be developed.

Note that, in general, there is considerable agreement that cost/benefit analyses do not yield decisions! Analysis can improve the likelihood that a judgment or decision is a good one but its principal role is to sharpen the intuition and judgment of the decision-maker.

C. THE COST BENEFIT ANALYST

It is also important to understand that the "analyst"² is not, and cannot reasonably be expected to be the decision-maker. Rothenberg points to potentially serious ambiguity in the role of the analyst thusly,³ "often he (the analyst) is an outside consultant, asked to perform evaluation for a specific agency of a specific governmental jurisdiction. The agency sees its responsibilities in ways which often differ from the consultant's perception. The latter is likely to see the agency's mission as subsumed within a larger one, larger both with respect to that jurisdiction and to more encompassing jurisdictions. His definition of the relevant population (the population for whom changes in well-being are being considered) will generally differ from that of his client with important policy implications. Yet if he acts upon it he is likely to render his advice unacceptable

¹ U.S., GAO, The Comptroller General, Impartial Cost-Effectiveness Studies, p. 8

² Unless otherwise noted all references in this chapter to the "analyst" refer to the person responsible for conducting the cost/benefit analyses.

³ Rothenberg, J. "Cost Benefit Analysis: A Methodological Exposition" in Handbook of Evaluation Research, Vol. 2, 1975, p. 75

to the client. If he deliberately adopts the vantage of his client, he may do real violence to his conception of the evaluative problem. Much of the same quandry is present with regard to the question of the relevant alternatives, where the client agency is likely to feel that its options are considerably more circumscribed than does its consultant analyst". In short, the expertise of the analyst is in the evaluative process itself, not in the system, or issue, or entity being subjected to evaluation. Obviously, the particulars of the analyst's role, a vitally important one, require clarification.

The analyst, charged with the responsibility for evaluating error-remediation alternatives for a military tactical ADP system serves as a catalyst, a coordinator, and as a "process" leader. He provides guidance in terms of the logistics, techniques, procedures, forms, etc. that are required during the course of information-gathering to insure that the information obtained is in a form compatible with the decision-maker's needs and objectives. At each stage of information-gathering the analyst should instruct the participants in the process as to methods, particularly those that relate to dealing with uncertainty and divergent (with respect to participant's) inputs. A critical task of the analyst is to determine who [organization, sub-unit, or individual(s)] provides what information. Human nature dictates that solutions, like beauty, are in the eyes of the beholder; thus, hardware people "see" hardware solutions, software folks think software alternatives, etc. From the point of view of the analysis process this natural proclivity approximates an ideal and represents a working goal for the analyst. What he does not want is the "tactical value of information" emanating from the hardware man, or alternatively, "design options for storage buffers or interface ports" being provided by the field commander. That is not to say that expertise in one area makes the participant/specialist blissfully ignorant of all other areas; it is simply a matter of focusing considerations of the participating experts in their areas of expertise, and enhancing the probability that such information is credible. If and when a participant offers comment, judgment, or specific data outside his bailiwick, it is incumbent on the analyst to insure a review of the information by the appropriate experts.

In this brief treatment of the role of the analyst only the highlights of his contribution have been discussed. In practice a considerable amount of diligent, probing, time-consuming investigation is involved. His task is completed when, for a given specific objective or set of objectives he has presented the cost-benefit solution space to the decision-maker.

D. THE SOLUTION SPACE

For the sake of clarity, that solution space, the end-product of a cost/benefit analysis (the point of departure for the decision-maker) can be characterized graphically.

Figure 7, for example, shows a typical solution space containing "G" alternative remediation procedures; the location of each alternative represents the benefits and costs that would accrue to each alternative should it be selected as a "best-fix". Hypothetical limitations have been imposed on this sketch via a vertical line (a given value on the abscissa) to represent a maximum allowable cost and a horizontal line (a given value on the ordinate) to represent a minimum performance requirement. It should be recognized that this representation can be extended or expanded to accommodate the range of considerations pertinent to either of the coordinate dimensions. Benefits, for example, can be expressed in many forms -- report accuracy, comfort, consistency, timeliness, etc., etc.; costs can be dollarized negative benefits, capital outlay, operating or maintenance costs, other intangibles, etc. etc.; finally, combinations of either costs or benefits or both are possible. In a similar fashion, each solution may contain one or more elements (e.g., one hardware modification, two hardware modifications; or a hardware and a software improvement) and each solution space may refer to a singular error, class of errors, or all errors generic to a specific system configuration. In addition, the hypothetical limits (maximum cost/minimum benefits) may be based upon a single dimension or a combination of dimensions. Since the customary use of "maximum" vs. "minimum" in their relationships to "costs" and "benefits", respectively, has been inverted, it seems prudent to offer a brief explanation before proceeding to the mechanics of the analysis that will lead us to the solution space desired. Normally, it is desirable to maximize benefits and minimize costs -- that is ultimately what is intended here. Use of the inverted limits, however, illustrates the ability of the decision-maker at the outset to eliminate alternatives which, although plausible, fail to qualify for further consideration because they do not meet either external or internal specifications that are more or less cast in concrete. (The system must provide "at least" ... or the budget for this item is "no more than X dollars"). The ambivalence implied in the "concreteness" of those limitations or constraints attends to the level of authority and/or flexibility vested in the decision maker.

For example, if an alternative, say "X", yields three times the benefits of the next-best alternative but its cost is

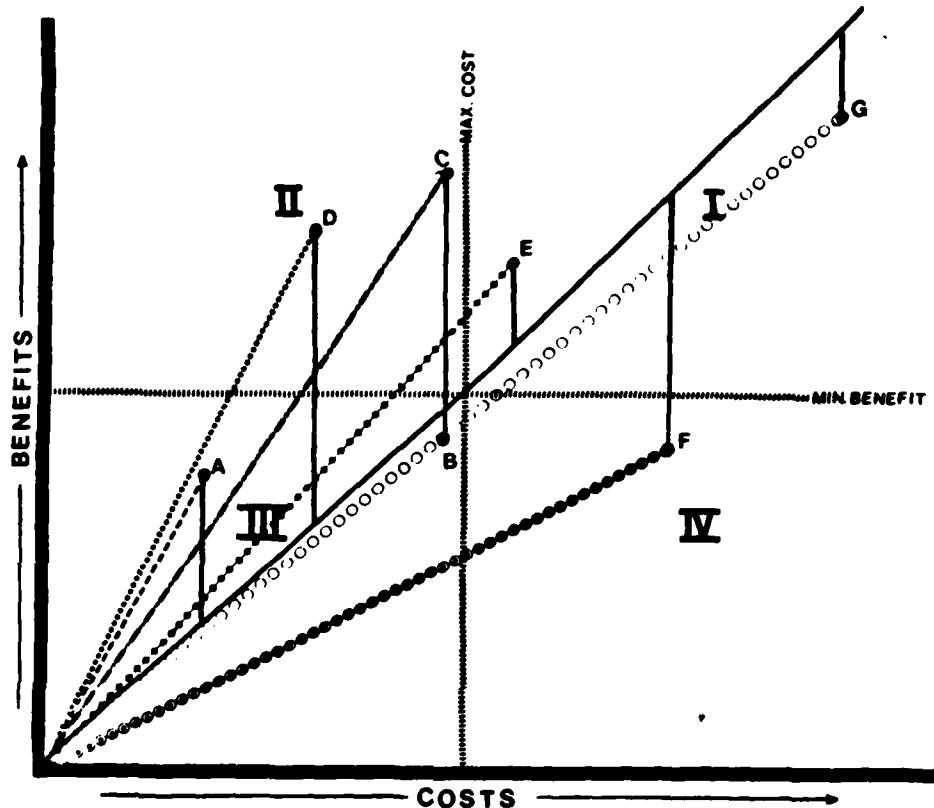


Figure 7. Typical Solution Space Showing Relative Locations of Hypothetical Alternative Remediations (A thru G)

- 1) To satisfy minimum performance requirement(s) (minimum benefits) - only those alternatives that fall in quadrants I and II can be considered.
- 2) With a budget limitation imposed -- only those alternatives that lie in quadrants II and III can be considered.
- 3) If the objective were to maximize the benefit within a given cost limitation -- alternative "C" would be the choice.
- 4) To maximize benefits regardless of cost -- alternative "G" would be selected.
- 5) To make some improvements at minimum cost -- alternative "A" -- an unlikely rule.
- 6) To meet the minimum requirements specifications for minimum cost -- alternative "D".
- 7) To make a "best buy" decision -- to select that alternative which yields the greatest number of benefits per unit cost ...

nominally a shade over budget -- should the decision-maker dismiss the "X" option because it fails the first hurdle? Or should he retain it in the hopes that he can obtain a budget adjustment? In practice he may or may not have a choice!

E. OBJECTIVES AND SUBOPTIMIZATION

Any rational evaluation or decision process is based upon the attainment of some desired objective(s). In the context of choosing among several alternatives to remediate input errors, the apparent simplicity of that understanding is challenged. Generally stated the *raison d'être* for military ADP systems could, for example, be construed as "national security". Although few will disagree with desirability of the "preservation of national security" as an objective, its relevant dimensions can be frightfully broad. Although a utility function may exist between a particular remediation alternative, for a particular class of input error, for a specific system configuration, and the achievement of national security, it is virtually impossible for the analyst to derive its exact form. Thus, depending upon the clarity of, and the "level" at which the objective is stated, the analyst or the decision-maker could find himself in the exceedingly difficult, or more frequently impossible position of attempting to relate how well the various alternatives satisfy the stated higher-level objectives. To circumvent this difficulty the analyst must resort to "suboptimization"; that is, he must attempt to reformulate this broad higher-order objective into new objectives whose attainment is (1) more easily calculated, and, (2) is an indication of the attainment of the higher-order objective. It should be recognized that the reformulation process may be, and often is, an iterative one where objectives are successively redefined until goals amenable to measurement emerge.

Fox and Haney¹ emphasize the importance of the meaningfulness of the relationship between higher and lower objectives thusly: "If lower measures of effectiveness are used, it is important that the analyst recognize a relationship between those measures and the higher measure of effectiveness although it may not be possible to define the relationship in a precise quantitative fashion. The analyst should describe -- at least in general terms -- how an increase in effectiveness using the lower measure as a standard would result in an increase in the higher measure.

¹Fox, P. D., and Haney, D. G., "Some Topics Relating to Military Cost-Effectiveness Analysis." A paper presented to the Operations Research Society of America (May 1966).

The analyst should also determine, as explicitly as possible, disparities between the higher and lower measures of effectiveness."

Hitch addresses the relationship in a slightly different way:¹ "The criterion for 'good' criteria ... is always consistency (emphasis added) with a 'good criterion at a higher level'. Hitch warns, however, that necessary connections between criterion levels may not exist and that either (1) effects must be assessed at the next higher level, or (2) where the analyst is aware of inconsistency between the suboptimization criterion and a higher-level criterion, allowances must be made for gains or losses imposed on other operations related to the higher level criterion.

F. COSTS

Whether or not the analyst is engaged in costing for systems improvement or system design (future applications), an initial distinction must be made as to whether or not the system under study is considered as an interacting component in the framework of the total military force, or as a stand-alone system. The former, "total-force-structure costing", is much more laborious than the latter, "individual-system costing" since it must attend to all interactions among individual force-component systems. Given the limitations on scope to the input interface of tactical military systems, any proposed evaluation exercise should be restricted to the individual-system type of costing.

For military systems, three broad cost classifications are recognized: (1) Research and Development costs, (2) Initial Investment Costs, and (3) Operating Costs.

This cost classification schema corresponds roughly to the time frame in which costs are incurred. Each solution alternative, whether it be remedial or initial system design, must be examined, vis-à-vis the elements of cost in each category. A typical listing of the costs included in each category adapted from the Army's Engineering Design Handbook² is as follows:

¹ Hitch, C., "Sub-Optimization in Operations Problems." J. Operations Research Society of America, Vol. 1, No. 3 (May 1953) pp. 87-99

² AMCP 706-191, "Engineering Design Handbook", pp. 245-246

Research and Development Costs

● Design and development

Preliminary research and design studies
Development engineering and hardware fabrication
Development instrumentation
Captive test operations
Facilities

● System Test

Test-vehicle fabrication
Vehicle spares
Test operations
Test ground support equipment
Test facilities
Test instrumentation
Data reduction and analysis
Maintenance, supply, miscellaneous

● System management and technical direction

Initial Investment Costs

● Installations

Construction of facilities

● Equipment

Primary-mission equipment
Specialized equipment
Other equipment

● Stocks

Initial allowances
Maintenance float
Equipment spares and spare parts
Consumption stocks

● Initial Training

● Miscellaneous investment

Initial transportation of equipment and spares
Initial travel

Operating Costs

- Equipment and installations replacement
 - Primary-mission equipment
 - Specialized equipment
 - Other equipment
 - Installations
- Training
- Pay and allowances
- Services and miscellaneous
 - Transportation
 - Travel
 - Other services and miscellaneous
- Indirect administrative and supportive costs

Certain of these cost elements may, of course, have little apparent relevance to specific remediation solutions; e.g. an alternative which specifies upgrading the level of operator personnel may require consideration of initial training on the system and certain miscellaneous investment costs but would have little impact on other areas save for costs savings. Presumably, the higher the level of the operator, the more efficient would be his performance, hence, the consumption rate of consumable stocks and supplies as well as time would decrease (in other words, this latter cost can be construed as having positive as well as negative utility).

The requirements of a specific system will dictate the extent to which incremental costing is appropriate. Incremental costing accounts for the additional costs associated with additional effectiveness and includes current assets, sunk costs, and salvage value. Current assets refer to the personnel, equipment, and facilities of the existing system and the extent to which they satisfy requirements of each of the competing alternatives. To the extent that existing assets can be utilized, they bear no additional costs. Sunk costs represent expenditures made in the past and inasmuch as sunk cost is the same for all alternatives, there is no need to consider it in the analysis. Salvage value refers either to the estimated scrap value of the system (or the remediation solution) when the useful life of the system has expired, or the cost savings that might be realized through a transfer or sale to another organization at that same point in time.

Amortization and discounting, two time-related costing elements, are not normally considered in military or government investments. Basically, amortization amounts to depreciating both the R&D and Initial Investment costs over the useful life of the improvement (or the system). Discounting, or the cost of lost opportunity that would otherwise accrue to present money, is a highly controversial issue, particularly when military investments are under scrutiny. The handbook supports this general position but quotes a discounting rate of 15 percent for use in Army studies (presumably, for those rare instances where the discounting of military investments is deemed relevant).

Like every other component, factor, or technique considered in cost/benefit applications, cost estimates (whether from catalog prices, cost-estimating relationships, or an estimate based upon a similar system) contain a great deal of uncertainty about them. Cost-estimating uncertainty derives in part from statistical errors in the cost data with more serious errors stemming from the assumptions and relationships that provide the basis for the cost estimates. Methods for dealing with uncertainty include contingency analyses, sensitivity analyses, *& fortiori* analyses and the like. It should be noted that a large variety of computer programs (for estimating logistics, life cycle and system development costs, operating costs, system support and maintenance costs, etc.) are available.¹ Most employ parametric analyses in an interactive mode that enables the analyst to use "what-if" strategies in arriving at his estimations. The Army's GEMM (Generalized Electronics Maintenance Model) which partitions maintenance costs to the piece-part level of detail and includes an optimum repair level analysis function is one such program. PRICE, another program, developed by RCA is suitable for both hardware and software cost estimation.

G. BENEFITS

Although intuitively easier to conceptualize than cost-estimation (i.e. a benefit is that which is good, has effectiveness, value, worth, or utility) the measurement of benefits is often a more difficult and at times impossible task. Indeed, benefit-estimation has as much or more uncertainty surrounding it than does cost estimation. How, for example, can the analyst measure "increased national security"? Obviously, he cannot and if he cannot then a value cannot be placed on the benefit. The

¹"Computers Analyze Cost-Effectiveness," Aviation Week and Space Technology, January 29, 1979, p. 194

importance that was attached earlier in this chapter to the specification of goals and objectives bears recalling. If objectives are not precisely stated, then a suitable measure of their attainment will not be possible. This does not mean that every benefit can be quantified. Kazanowski notes what he calls the "quantification fallacy" in which an assumption is made that every dimension of importance relevant to the decision can be quantified.¹ Certain factors no matter how diligently approached by the analyst simply elude quantification. There is considerable disagreement over the manner in which non-economic benefits (peace of mind, security, etc.) should be addressed. Quasi-or shadow-prices, the assignment of prices that would represent a market value if a market place existed, are frequently suggested and as frequently dismissed. Opponents of shadow pricing argue that the analyst should look elsewhere to ascertain values for the benefits in question. In the area of benefit estimation, uncertainty in general and unquantifiables in particular, are usually resolved via consultation with experts knowledgeable in a particular field of interest. The specific methods and procedures are, like the procedures for handling cost uncertainties, well known and generally available. These procedures vary from consultation with a single expert, to consultations with several on an individual basis, to face-to-face confrontations between a number of experts with and without interaction and discussion. Perhaps the best known of these methods is "The Delphi Technique".² The Delphi Technique attempts to achieve a consensus opinion from a panel of experts in ways that avoid face-to-face confrontation and maintain anonymity. The process is an iterative one that solicits information (judgments and opinions) usually via questionnaires. Information is collected, and the range of responses is assessed by the analyst and a copy of this information returned to the experts (preserving anonymity). The experts are asked to revise their opinions in light of the feedback, with "reasons" being attached to those opinions which were lower than the first or higher than the third quartile. The reasons and the revised opinions are again submitted to the experts and on this round the experts are asked to evaluate the reasons and then revise their estimates. If the revised estimates still fall outside the quartile criteria, the respondent(s) providing the

¹ Kazanowski, A. D., "Cost-Effectiveness Fallacies and Misconceptions Revisited." A paper presented to the Operations Research Society of America (May 1966).

² Helmer, O., "Convergence of Expert Consensus Through Feedback." The RAND Corp., P-2973 (Sept. 1964).

out-of-range answers are asked why he or they were unconvinced by the argument that would have brought the response closer to the median. A new range is calculated and sent back to the experts along with the new arguments and a final opinion is solicited. At each iteration (and there may be as many as time and resources permit) the dispersion in responses (disagreement among experts) is generally less than it was on the preceding iteration.

The median of the final estimates is used as an estimate of the value sought. Note that the procedure is useful not only to derive specific values on a given dimension of worth or utility, but can be used equally well earlier in the cost/benefit process to establish objectives and/or the dimensions of importance against which every possible outcome of each remediation alternative should be evaluated.

H. MULTI-ATTRIBUTE UTILITY MEASUREMENT

The discussion so far has focused upon general factors which must be considered in any serious cost/benefit analysis (viz. objectives, costs, benefits, criteria, etc.) and certain tools and techniques (suboptimization, Delphi, sensitivity) which are useful in the reduction of uncertainty and/or the treatment of disagreement among and between experts. What is lacking thus far, is a procedure for blending these considerations to produce the desired solution space. In order to achieve a very clear perspective of the relationships between the factors, techniques, and procedures (to be discussed) one need only consider an analogous task -- baking bread. The factors in the cost/benefit analysis are represented by the ingredients -- the flour, sugar, eggs, water, etc.; the techniques, by kneading, rolling, and rising; and finally, the procedure for blending by the recipe itself -- the prescription which details the "what", "when", and "how" in typical cookbook fashion, i.e., a step-by-step approach.

Multi-Attribute Utility Measurement (MAUM), a decision-theoretic evaluation procedure, championed by Edwards et al¹ is one such recipe. The essence of MAUM as described by the authors is flexibility in combining quantitative evidence from different sources, different lines of inquiry, and different techniques of investigation. This essence has been characterized as convergent validity; the more different lines of evidence which point to a particular conclusion, the more confidence the analyst or decision-maker will have in that conclusion.

¹Edwards, W., Guttentag, M., and Snapper, K. "A Decision Theoretic Approach to Evaluation Research" in Handbook of Evaluation Research, Vol. 1, 1975.

It is important to recognize that in applying the utility analysis to a choice among remediation alternatives (e.g. "to introduce an interactive dialogue with smart terminals" vs. "to upgrade the level of user personnel" vs. "to upgrade the training of user personnel") we are conducting the analysis solely to choose that action or remediation alternative which maximizes utility, however, it is the outcomes of the actions, not the actions themselves which have utility attached to them.

Outcomes can be useful to different persons for a number of very different reasons, that is they have value on a number of different dimensions. Consider, for example, a keyboard input error which results in erroneous coordinates for target location. A number of remediation solutions exist to rectify the problem. A correct system-generated report (an outcome) has obvious utility for the field commander in his tactical or strategic plans; however, if the report arrives several hours late it could be not only useless but disastrous. Thus, at least two value dimensions, accuracy and timeliness of reporting, have worth, value, or utility to the field commander. Note that the field commander might not attend at all to the mechanics of report production. He cares not how the report is produced, what or how many activities went into its production, or the complexity, or the cost. He simply wants accurate and timely information. The unit commander of the data processing facility on the other hand, might consider the field commander's value dimensions as important because of the potential consequences in the field but in addition be concerned with the quality of his organization's products (reports) because they impact on his unit's efficiency and effectiveness ratings (a value dimension). More important at this level, however, is the consideration of outcomes that accrue to the different alternatives. If a people "fix" is involved, the ADP unit commander would almost certainly be concerned with (attach value to) personnel availability, questioning perhaps whether more or less staffing is required; whether the staffing requirement could be physically accommodated in the work space; if re-training of the incumbent crew is indicated, what provisions can be made to maintain unit readiness in the time-frame of the training, etc. He would also be concerned, but possibly to lesser degree, with outcomes occasioned by a hardware "fix" -- downtime would almost certainly be of negative value -- how negative depending upon how long it would take to effect a component modification. Without belaboring the point further, suffice it to say that any of the several outcomes that result from a particular action, have utility (positive or negative) on a number of different value dimensions to a number of different people. To employ the MAUM procedure each outcome is located on each dimension of value.

Location measures are then combined through an aggregation rule -- usually a simple weighted linear combination, where the weights refer to the relative importance of each dimension of value. Derivation of the weights is almost always accomplished through pooled expert judgments.

It would be difficult to improve on the synoptic presentation offered in the Handbook of Evaluation Research, of the steps involved in the MAUM process. Indeed, social scientists are indebted to the trio responsible -- Edwards, Guttentag, and Snapper -- for the clarity of their exposition.¹ That presentation contains an excellent discussion laced with concrete examples (of the application of the process and its principles) in a level of detail that unfortunately, for purpose of this report, precludes repetition. An earlier paper, however, by Guttentag and Snapper² provides an abbreviated version of the steps involved in the process. These are:

- "Step 1: Identify the organization whose utilities are to be maximized.
- Step 2: Identify the issue or issues to which the utilities needed are relevant.
- Step 3: Identify the entities to be evaluated.
- Step 4: Identify the relevant dimensions of value.
- Step 5: Rank the dimensions in order of importance.
- Step 6: Rate dimensions in importance, preserving ratios.
- Step 7: Sum the importance weights, divide each by the sum, and multiply by 100.
- Step 8: Measure the location of the entity being evaluated on each dimension.

¹ Edwards et al., "Decision Theoretic Approach," I, pp. 153-157

² Guttentag, M., and Snapper, K. "Plans, Evaluations, and Decisions," Evaluation, Vol. 2, January 1974, pp. 58

Step 9: Calculate utilities for entities. The equation is

$$U_i = \sum_j w_j u_{ij}$$

remembering that

$$\sum_j w_j = 100$$

U is the aggregate utility for the i th entry; w_j is the normalized importance weight of the j th dimension; u_{ij} is the rescaled position of the i th entity on the j th dimension. Thus w_j emerges from Step 7 and u_{ij} emerges from Step 8.

Step 10: If a single act is to be chosen, the rule is simple: maximize U_i . If a subset of i is to be chosen, then the subset for which $\sum_i U_i$ is a maximum is best."

VI. APPLICATION OF THE MAUM PROCESS TO THE TOS
INPUT ERROR/REMEDIATION DECISION PROBLEM:
AN ILLUSTRATIVE EXAMPLE

The Multi-Attribute Utility Measurement process described in the preceding section of this report is a version that is oriented toward easy communication and use in complex environments where there is a premium on the time and availability of decision-makers. It is not intended as a substitute for the careful, painstaking, cost/benefit procedures contained in AMCP 706-191; indeed, consistent with its basic dependence on convergent validity, MAUM accepts any reliable evidence that is obtainable -- thus, it welcomes (on an as-available basis) data generated by more formal techniques such as cost-sensitivity analyses, inventory and replacement, queuing theory, etc. What it offers to decision-makers, that the mathematically complex cost/benefit procedures lack, is a method that is psychologically meaningful -- an important contribution to decision-makers who are expected to render judgments that are intuitively reasonable.

It will soon become clear that the illustrative application to TOS is just that -- illustrative. At each step in the process plausible assumptions and results are presented simply to illustrate the thinking involved in that step and to typify what might be the relevant conclusions drawn from that same step.

Obviously, the real dimensions of importance and values to those concerned with the TOS facility will not be known until the MAUM procedure is directly applied to the TOS input error/ remediation problem. If and when that application occurs, there is ample reason to expect that the evaluation and choice(s) will be very close to those that would have obtained from the extremely time-consuming, more formal procedures required by AMCP 706-191, and that answers will be available in a more timely manner -- perhaps "orders-of-magnitude" sooner.

STEP 1: *Identify the person or organization whose utilities are to be maximized.*

It is extremely important for the analyst to understand the mission of the tactical information system, its place in the organizational structure of the Army and the nature of its interfaces with other support and line activities. Only then will he be able to assess the extent to which a proposed system remediation has the potential for impact outside of the immediate system environment. To the extent that outcomes (of alternative remedial actions) have effects on interacting organizations or

units, these organizations have a stake in the decision -- they should, therefore, have a corresponding voice in the decision process. Personnel who are able to represent these units must be identified and induced to cooperate in the utility evaluation.

Obviously, the ADP unit commander and his subordinates, the operators of the system, are the first to be considered. Outside of the immediate system environment, those organizations that interact with, are impacted by, or which impact on the ADP system must be represented. Users of the reports produced by the system (for example, HQ, Intel, or Combat units) must be represented. Different echelons of command, will have varying degrees of dependence on the system (hence, differential utility functions, though inexact, will exist) and these differences as well as people qualified to speak to them should be identified.

The choice of a remediation alternative will likewise bear some utility or disutility to the systems development activity that services and supports the ADP facility. Hardware alternatives, or remediation solutions containing equipment development, fabrication, or procurement will almost certainly impact, for example, on the systems engineering and maintenance activities that support the tactical information system in question. Systems analysts will be concerned with the effect that either a hardware or software modification will have on the systems availability, dependability, and capability.

Suffice it to say at this point that the participants in the evaluation process should include those for whom the change "makes a difference". Unfortunately, there exists no convenient formula for determining how many participants are enough. Thus, the determination of how many voices and which specific voices need to be heard should be guided in large part, by the number of direct information dissemination channels to units outside the TOS proper, and the number of major lines of support to the system.

STEP 2: *Identify the issue or issues (i.e. decisions) to which the utilities needed are relevant.*

Edwards et al. assert that utility is a joint function of the evaluator,¹ that which is being evaluated, and the purpose for which the evaluation is being conducted; he and his colleagues

¹ Edwards, W., Guttentag, M., and Snapper, K. "A Decision Theoretic Approach to Evaluation Research" in Handbook of Evaluation Research, Vol. 1, 1975.

suggest that the last argument of that function, purpose, is often neglected. In general, the purpose for the overall trade-off exercise can be identified as "to select a remediation solution(s) which will eliminate or reduce input errors". It is clear, however, given the specificity and nature of the error-classification scheme discussed earlier that certain input errors occur in a mutually exclusive fashion with respect to other errors. Consider, for example, errors of commission and omission -- a "fix" contemplated for an error of commission has limited impact on existing or potential errors of omission whereas the obverse situation -- correcting errors of omission -- may indeed, induce errors of commission! Which is the more important typology in terms of error consequence, severity, pervasiveness, and frequency, etc. to users of the information or to producers of the information? Is the question important, or in other words, is the issue relevant to the decision process? Given the real-world costs of bad vs. missing information it would seem that certain varieties of error should undoubtedly have a higher priority of remediation than others.

That the participants would almost certainly identify "error priorities" as a relevant issue can be supported by another example: An enemy squadron misidentified in a computer-generated report as a squad (perhaps the result of an abbreviation error) carries with it the potential for a serious tactical disadvantage should a field commander elect to engage the enemy force with what he believes to be a superior force (a platoon). An extreme example perhaps, but it serves to illustrate the point.

The enumeration of issues which are relevant (for example, the establishment of error remediation priorities, and the selection of remediation alternatives for a given error or class of errors) stems directly from a detailed consideration of the evaluation objectives. It will be recalled that one technique that is almost always employed to reduce lofty objectives to measurable ones is suboptimization. In practice the suboptimization technique might be applied by the analyst working in consultation with the ADP unit commander since the initial goal -- to remediate input errors -- is somewhat narrowly confined (where multiple goals are involved, group processes or discussions might be required to enumerate and define several relevant issues). Generally, however, groups of expert participants such as those identified in Step 1 do not become involved until later in the evaluation (at Step 4).

Note that the line of reasoning pursued to this point has resulted in at least two purposes for which the evaluation is being conducted:

Issue #1: The prioritization of errors for remediation
and

Issue #2: The selection of a remediation alternative
for a specific error type or class of errors.

There is no compelling argument that requires that all participants evaluate all issues but there is, however, an implicit ordering in the treatment of issues. That is to say that those participants who are TOS personnel or users of TOS reports should evaluate the first issue (the relative seriousness, consequence, frequency of errors) while TOS personnel and support activities to TOS should concentrate on the impact of system changes necessitated by the several remediation alternatives, given the resolution of the first issue. So, in effect, the evaluation process is staged -- Issue #1 comes first, and only then, when remediation priority has been established for errors, solutions for those errors can be appraised. Since for any number of issues the process steps are identical, the staging of the evaluation process will not be specifically referenced in the balance of this discussion.

STEP 3. *Identify the entities to be evaluated.*

The entities to be evaluated are, in short, the errors (or error types) and the remediation alternatives themselves. This simple truth is offered despite the earlier assertion that outcomes not acts have value attached to them. Since it is always necessary to "draw the line" somewhere, there comes a point where it is necessary to stop considering outcomes as "opportunities for further decision" (the situation implied is analogous to the persistent "why" which a child asks each time his preceding "why" is answered). At this point, which is determined solely by convenience, the action is treated as having intrinsic value and is, in effect, an outcome. Edwards et al explains that "This amounts to treating the action as having an inevitable outcome, that is, of assuming that uncertainty about outcomes is not involved in the evaluation of that action".¹

¹ Edwards, W., Guttentag, M., and Snapper, K. "A Decision Theoretic Approach to Evaluation Research" in Handbook of Evaluation Research, Vol. 1, 1975.

Thus, the entities or outcomes to be considered are of two types: errors and the specific remediation solutions addressed earlier, as well as any that might emerge as the evaluation progresses through the mutual exchange of ideas and discussions between participants. A brief summary listing of errors relevant to issue #1 and the remediation alternatives relevant to issue #2 is presented here to dispel any confusion as to what is being evaluated:

Entities (for Errors):

Omission of strength information for friendly forces
Omission of locations of friendly forces
Omission of subordinate unit tasking or missions of friendly forces
Errors of commission in strength information of friendly forces
Errors of commission in location of friendly forces
Errors of commission in subordinate unit tasking or missions of friendly forces
Timeliness (within 15 minutes of suspense time) of strength information for friendly forces
Timeliness (within 15 minutes of suspense time) of location information of friendly forces
Timeliness (within 15 minutes of suspense time) of subordinate unit tasking of friendly forces
Timeliness (within 15 minutes of suspense time) of information about enemy intentions
Timeliness (within 15 minutes of suspense time) of enemy unit location information
Timeliness (within 15 minutes of suspense time) of enemy unit strength information
Omissions in enemy unit strength information
Errors of commissions in enemy unit strength information
Omissions in enemy unit location data
Errors of commission in enemy unit location data
Omissions in data on enemy intentions
Errors of commission in data on enemy intentions
•
•
•

Entities (for Remediation Alternatives):

Broad Remediation Strategies
Personnel Selection Procedures
Training Methods
Processors
Memory
Dialogue Characteristics

Specific Remediation Strategies

Replace MIOD with AN/UYQ-19

Provide 2 weeks OJT after current classroom course

Add 5 megabytes disc storage

-
-
-

Note that the list of alternative remediations should include all solution elements and combinations of elements which appear to have particular promise. In this way, the individual contributions of single solutions can be assessed in a tradeoff with the additive effects of other solutions. For example, it might be desirable to evaluate the introduction of smart terminals with local edit and validation, with and without interactive dialogue and with and without validation using information from a central summary data base.

STEP 4: *Identify the relevant dimensions of value*

The first three steps dealt with 1) whose utilities were to be maximized, 2) for what purpose(s) the utilities were to be maximized, and 3) what utilities were to be maximized. In Step 4, the focus is on determining the criteria, or dimensions of value, that the participants deem to be important in evaluating remediation alternatives. In effect, they represent the goals that each participant would like to see achieved, not stated in terms of absolute numbers but in terms of the dimensions of importance. A simple example will demonstrate this difference: The appropriate statement of the overall evaluation goal at this step is "to reduce the occurrence of input error" not "to reduce input errors by 75%".

There are, of course, a great number of specific value dimensions that can be generated. The ADP unit commander will easily generate several for a single alternative remediation procedure; he may find that he applies different criterion dimensions to the consideration of other alternative solutions. Given that the same opportunity to list the dimensions of value (the criterion dimensions) is available to as many participants as there are identified in Step 1, a caution must be provided: to list too many dimensions will create unnecessary scaling and weighting difficulties later in the process. At this point then, gross dimensions are fine -- in fact they are preferred. Participants should be encouraged to generate a listing of their

preferred goals and to then eliminate the less important ones. They might also be reminded that the list of dimensions will be subjected to revision later and that the result of that revision will most likely expand rather than reduce the number of dimensions.

An illustrative listing of the kinds of dimensions which might be identified is as follows:

- Extent to which error contributes to mission degradation where assigned mission is moving to attack
- Extent to which error contributes to mission degradation where assigned mission is coordinated attack
- Extent to which error contributes to mission degradation where assigned mission is air defense
- Extent to which error contributes to mission degradation where assigned mission is delay action
- Extent to which error contributes to mission degradation where assigned mission is counter attack
-
-
-

Dimensions of Value for Issue of Selection of Remediation Alternatives:

- To decrease system response time
- To promote system availability, dependability, capability
- To increase system interoperability
- To decrease user response time

- To decrease the frequency and length of queues of data to be input
- To reduce dependence on external support system
- To increase user confidence in the system
- To decrease size and weight of system components
- To decrease electronic signature
-
-
-

In practice the analyst might have to invoke one of the group processes such as the Delphi Technique in order to achieve a consensus on a manageable number of goals. Edwards et al. contend that 8 is plenty and 15 is too many. The task for the group action would be to restate or combine goals (if a hierarchy exists) to arrive at approximately 10 (For instance system response time is a dimension which is subordinated to system availability).

STEP 5: Rank the dimensions in order of importance

When all of the dimensions of value have been collected from the participants (via questionnaire, interview, etc.) a composite listing is generated by the analyst. The participants are then asked to rank the dimensions included on the list in order of importance, preferably using group processes (Delphi). Following the iterative procedure, when all arguments (pro and con) for the rank assignments have been heard and exchanged, separate final judgments of rank are solicited from each individual. Using these final rankings, the analyst forms a composite ordered listing based on the sum of the ranks assigned to each dimension across participants.

STEP 6: Rate dimensions in importance, preserving ratios.

In this step we are concerned with how much more important one dimension of value is compared to another (and, eventually all others); that is, we wish to derive relative weights for each of the dimensions of value. To accomplish the rating each participant is asked to examine the ranked list resulting from the previous step (assume that we are dealing with the value

dimensions for the selection of remediation alternatives, i.e., those pertinent to Issue #2), and assign to the least important dimension an "importance" of 10. The next-to-the-last dimension on the ranked list is then compared to the least important and the participant is required to judge how many times more important it is than the least important. Finally he must assign a number reflecting that ratio. Thus, if the next-to-the least important is three times more important than it would be assigned an "importance" of 30 (if there is judged to be no difference in importance, then a value of 10 is assigned). The participant is expected to continue up the list, judging each dimension in a relative fashion until all dimensions of value have been assigned an importance rating. As soon as the rating assignments are completed, the participant should review all pairs of dimensions to insure that his ratio assignments are consistent with each other; i.e., if, in his review he decides that a dimension with a rating of 60 is upon reexamination not really twice as important as the one with 30, he can change either or both to accommodate his revised opinion.

Once again individual differences are likely to yield a wide range of importance ratings for the given list of ranked dimensions but this time there is no need to resort to group processes to generate consensus.

STEP 7: Sum the importance weights, divide each by the sum and multiply by 100.

The instruction contained in the step title says it all. The object of this step is simply to produce for each criterion dimension (of value), a metric that is similar to a probability. Let us assume for purposes of illustration that only six dimensions survived the recombination exercise in step 4 and for convenience let us label them D1 through D6. The ranking exercise (step 5) rearranged the original list to yield an ordinal importance scale. Thus we might have:

D4	•
D2	•
D5	•
D1	•
D3	•
D6	•

The activity in step 6, the rating task, would have produced relative weights, thusly:

D4 = 150
D2 = 100
D5 = 80
D1 = 80
D3 = 40
D6 = 10

Now to avoid possible confusions with the next step we will depart slightly from the Edwards prescription and recommend that a value of 500 be used as the multiplier in normalizing the weights for the criterion dimensions. The choice of scale is an arbitrary one which, in effect, simply distributes 500 points over the set of six dimensions. Note that if there are great importance gaps between a large number of dimensions the less important dimensions will be assigned trivial weights. (Thus, the admonition at step 4 to "hold down" the number of dimensions to be considered.) Thus, at the end of step 6 we would have, pursuing the example above:

<u>Unranked Dimensions</u>	<u>Ranked</u>	<u>Rated</u>	<u>Weighted/Scaled</u>
D1	D4	150	163.0
D2	D2	100	108.7
D3	D5	80	87.0
D4	D1	80	87.0
D5	D3	40	43.5
D6	D6	<u>10</u>	<u>10.9</u>
		<u>$\Sigma=460$</u>	<u>$\Sigma=500$</u>

Step 8: Measure the location of each entity being evaluated on each dimension

It will be recalled from step 3 that depending on which issue ("error remediation priority" or "remediation alternative selection") is being evaluated, the entities are either "errors or error classes" or "specific alternative solutions." Having established that simple fact, let us forget "entities" for the moment and turn our attention to the criterion dimensions. Basically, there are three classes of dimensions: those which are purely subjective, those which are partly subjective, and those which are purely objective. The classification scheme is necessitated by the fact that for certain dimensions, natural units of measurement exist and for others they do not.

"Timeliness" and "accuracy" can be measured, for example, in seconds and percentage of correct responses, respectively, but a natural scale for "ease of operation" or "comfort" is not readily available. To "locate" entities on dimensions, however, the dimensions must have scalar values; moreover, the dimension scales must be comparable.

For purely subjective dimensions the analyst identifies an appropriate participant/expert and instructs him to estimate the location of a particular entity (e.g. remediation alternative) using a 0 to 100 scale on the purely subjective dimension where 0 is defined as the minimum plausible value and 100 the maximum.

Partly subjective dimensions are those that have natural units but the participant/expert renders a subjective judgment as to the location of the remediation alternative on the dimension. Wholly objective dimensions, finally, are those for which "hard data" exists in natural units prior to the evaluation (recall that the procedure will accept any reliable evidence on an as available basis).

For the partly subjective and purely objective dimensions, minimum and maximum plausible values in the dimensions' natural units must be determined. For example, for a dimension that is specified "the extent to which the remediation alternative reduces errors of location," plausible natural limits could be ascertained by consulting current system performance records.

In natural units, the lower boundary of system-generated location errors would be zero for an error-free system, while current performance (on any suitable basis, no. of location errors/report; no. of location errors/month) would constitute the upper boundary for "reduction". In converting these to comparable scale units, 0 to 100, the natural scale and the transformed scale would bear an inverse relationship to each other, i.e. a value of 100 would refer to the error-free system and zero would be assigned to the natural unit representing current performance. Linear transformations are suitable for the scale conversion process, however, the transformation should not occur until the participants have "located" the entity on the dimension in its natural units.

STEP 9: Calculate utilities for entities.

In this step for each entity, we seek a weighted average importance across all dimensions. The formula $U_i = \sum_j w_{ij} u_{ij}$, is

the aggregate utility for the i th entity on the j th dimension (from step 8). To pursue still further our earlier example; w_j represents the scaled importance weights that result from step 7 ($D_4=163.0$; $D_2=108.7$; $D_3=87$; etc.), and U_{ij} represents the location of the i th entity on the j th dimension (from step 8). To be more specific, let us extend the example. Let us assume that the error prioritization evaluation has been completed and we now seek to select the "best" alternative remediation for a specific error. The "decision space" might resemble the partial table shown below:

		Remediation Alternatives					
		Hardware	Software	Personnel	Procedures	Composite Solution	
PRIORITY ERRO RS	e_1	a_1 a_2	a_3 a_4 a_5	a_6	a_7	a_8 a_9	
	e_2	a_1 a_2	a_3				a_4
	e_3	a_1		a_2			a_3
	e_4		a_1 a_2 a_3				a_4
	•						

The errors have been prioritized and the position of error e_1 means that there is more utility attached to its remediation than any other. But, the decision space indicates that there are nine alternative (a_1 through a_9) solutions for remediating e_1 . Continuing with our example in the language of MAUM, there are then, $i=9$ entities, which are located u_{ij} , on each of $j=6$ dimensions. The product of the location of the i th solution on the j th dimension is calculated and the resulting $j=6$ products are summed. The process is repeated for all $i=9$ solutions.

Since there was, in the previous step an insistent requirement for comparable scales, across dimensions, and step 7 produced a relative weighted "importance" for each dimension, equal

numerical differences between products ($w_j u_{ij}$) on different dimensions can be interpreted as equal changes in desirability.

To illustrate the equivalence, assume that location values for 7 of the solutions (a_3 through a_9) were zero across all dimensions and that the survivors, a_1 and a_2 had zero location values on all dimensions except for the two most important, D4 and D2), then let us further assume that:

a_1 has a location of 60 on D4

a_2 has a location of 30 on D4

a_1 has a location of 45 on D2

a_2 has a location of 90 on D2

Since, via step 7 it was determined that D4 is 1 1/2 times as important as D2 (i.e., 163.0/108.7) then,

$$U_{a_1} = 1.5 (60) + 45 = 135$$

and

$$U_{a_2} = 1.5 (30) + 90 = 135$$

or equivalently

$$1.5 (60-30) + (45-90) = 0$$

Note that this illustration puts the burden of final choice squarely on the decision maker because remediation solution a_1 and a_2 have identical aggregate utilities, hence equal desirability.

STEP 10: Decide

The solution space is, at the conclusion of step 9, determined. What remains for the decision-maker is the choice of which decision rule he is bound or elects to follow. Early in the previous chapter, a graphic representation of the typical cost/benefit solution space was provided. Accompanying that figure is a list of typical decision rules which could lead, given the same solution space, to very different choices depending upon which rule was employed.

One further point requires emphasis. Cost is always considered as a criterion dimension in the MAUM procedure when the purpose of the evaluation is to select the most cost beneficial entity.

In the event that a budget constraint is imposed (or, for that matter) any criterion dimension is constrained, then steps 4 through 10 should be ignored for the constrained dimension. C_1 , the cost estimate of a_1 , would then enter into the calculation of the benefit/cost ratio, U_1/C_1 . The choice, under budget constraints, then, would be to select alternatives in decreasing order of U_1/C_1 until the funds are exhausted. Lacking constraints, all dimensions including costs, are simply treated as are any other value dimensions.

Given that a verbalization of the decision process seems at times a bit involved, it would seem helpful to provide the reader with the "picture that is worth a thousand words". To this end, Figure 8, attempts to describe the steps involved, their flow, and interrelationships in a more graphic and, perhaps more understandable manner.

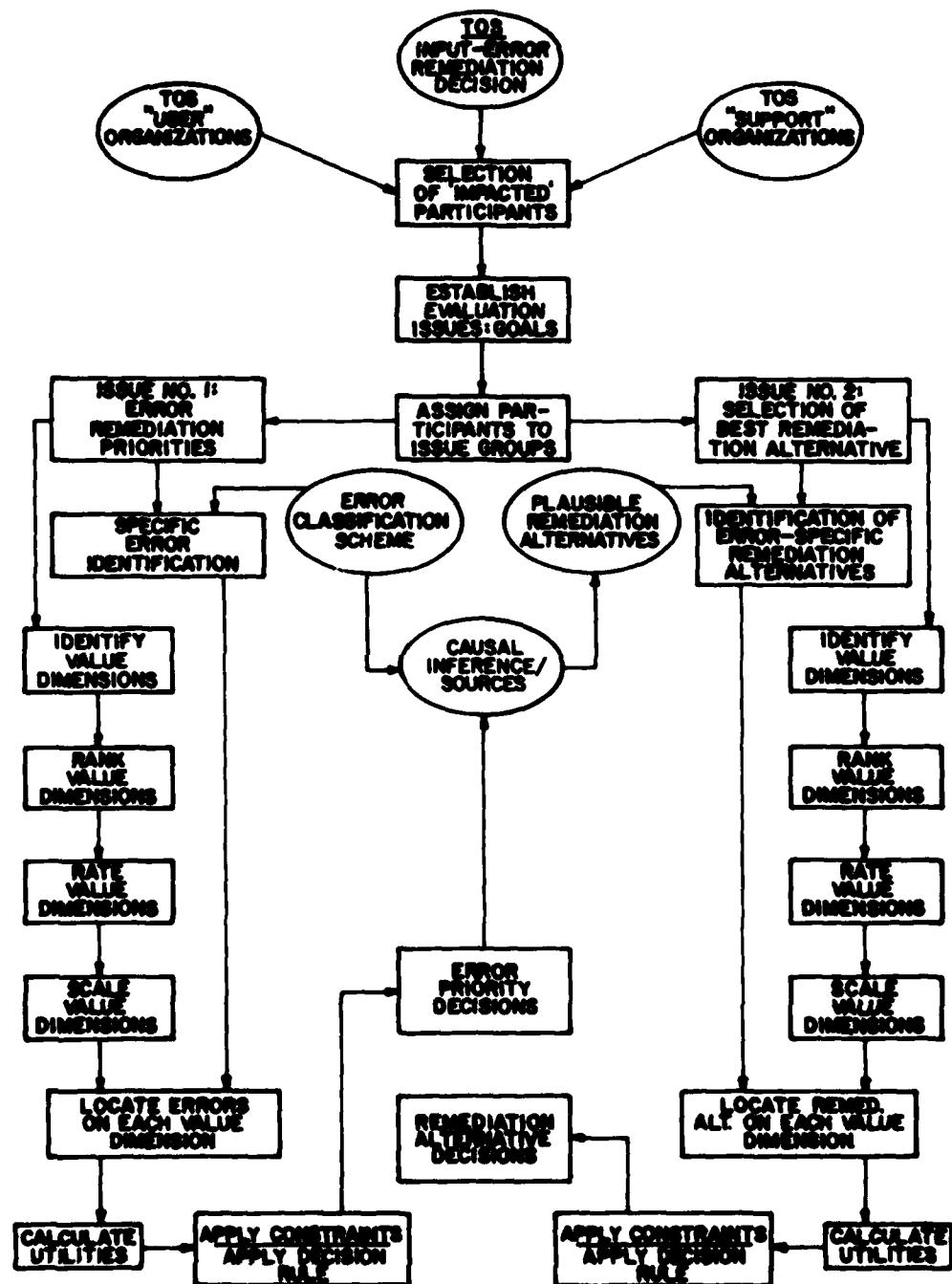


Figure 8. A Schematic Representation of the MAUM Decision Process Applied to the TOS Error Remediation Problem

REFERENCES

AMCP 706-191; Engineering Design Handbook: System Analysis and Cost-Effectiveness; U.S. Army Material Command, Headquarters, Washington, D.C., 9 April 1971.

"Computers Analyze Cost-Effectiveness," Aviation Week and Space Technology, January 29, 1979, p. 194.

Baker, J. D. "Acorns in Flower Pots/Psychologists in the Field": paper presented at the United States Army Human Factors Research Development Sixteenth Annual Conference, El Paso, Texas, October, 1970.

Bell, C. F. "Cost-Effectiveness Analysis as a Management Tool." The Rand Corporation, P-2988 (Oct. 1964).

Doherty, W. J. & Fischer, C. "Human Factors: Impact on Interactive Computing". A seminar presented January 16, 1979.

Edwards, W., Guttentag, M., and Snapper, K. "A Decision Theoretic Approach to Evaluation Research" in Handbook of Evaluation Research, Vol. 1, 1975.

Fox, P. D., and Haney, D. G., "Some Topics Relating to Military Cost-Effectiveness Analysis." A paper presented to the Operations Research Society of America (May 1966).

Gilb, T., & Weinberg, G. M. Humanized Input Techniques for Reliable Keyed Input. Winthrop Publishers, Inc. 1977.

Guttentag, M., and Snapper, K. "Plans, Evaluations, and Decisions," Handbook of Evaluation Research, Vol. 2, 1975.

Hammer, M. "Error Detection in Data Base Systems" National Computer Conference, 1976.

Helmer, O., "Convergence of Expert Consensus Through Feedback". The Rand Corp., P-2973 (Sept. 1964).

Hitch, C., "Sub-Optimization in Operations Problems." J. Operations Research Society of America, Vol. 1, No. 3 (May 1953) pp. 87-99.

Kazanowski, A. D., "Cost-Effectiveness Fallacies and Misconceptions Revisited." A paper presented to the Operations Research Society of America (May 1966).

Levine, R. A., "Systems Analysis in the War on Poverty." A paper presented to the Operations Research Society of America (May, 1966).

Martin, J. Design of Man-Computer Dialogues, Prentice-Hall, Inc.

Nawrocki, L. H., Strub, M. H., & Ross, M. C., "Error Categorization and Analysis in Man-Computer Communication Systems", IEEE Transactions on Reliability, Vol R-22, No. 3, August 1973.

Nickerson, R. S. "Man-Computer Interaction: A Challenge for Human Factors Research", Ergonomics, Vol. 12, No. 4, July 1969.

Pagano, A. M. and Sauerlander, O. H. "Benefit-Cost Analysis" In Roadway Delineation Systems, NCHRP Program Report 130, 1972, pp. 156-186.

Prest, A. R. and Turvey, R., "Cost-Benefit Analysis: A Survey." Economic J. Vol. 75, No. 300 (Dec. 1966).

Quade, E. S., "Cost-Effectiveness: An Introduction and Overview." The Rand Corp., P-3134 (May 1965).

Quade, E. S., "Some Comments on Cost-Effectiveness." The Rand Corp. P-3091 (March 1965).

Rothenberg, J. "Cost Benefit Analysis: A Methodological Exposition" in Handbook of Evaluation Research, Vol. 2, 1975.

Rouse, W. B. Design of man-computer interfaces for on-line interactive systems. Proceedings of the IEEE, 1975, 63.

U.S. General Accounting Office, The Comptroller General, Impartial Cost-Effectiveness Studies Found Essential to Selecting New Weapons; DOD B-163058 Report to the Congress, 21 August 1972.

DISTRIBUTION

ARI Distribution List

4 OASD (M&RA)
2 HQDA (DAMI-CSZ)
1 HQDA (DAPE-PBR)
1 HQDA (DAMA-ARI)
1 HQDA (DAPE-HRE-PO)
1 HQDA (SGRD-ID)
1 HQDA (DAMI-DOT-C)
1 HQDA (DAPC-PMZ-A)
1 HQDA (DACH-PPZ-A)
1 HQDA (DAPE-HRE)
1 HQDA (DAPE-MPO-C)
1 HQDA (DAPE-DW)
1 HQDA (DAPE-HRL)
1 HQDA (DAPE-CPS)
1 HQDA (DAFD-MFA)
1 HQDA (DARD-ARS-P)
1 HQDA (DAPC-PAS-A)
1 HQDA (DUSA-OR)
1 HQDA (DAMO-ROR)
1 HQDA (DASG)
1 HQDA (DA10-PI)
1 Chief, Consult Div (DA-OTSG), Adelphi, MD
1 Mil Asst. Hum Res, ODDR&E, OAD (E&LS)
1 HQ USARAL, APO Seattle, ATTN: ARAGP-R
1 HQ First Army, ATTN: AFKA-OI TI
2 HQ Fifth Army, Ft Sam Houston
1 Dir, Army Stf Studies Ofc, ATTN: OAVCSA (DSP)
1 Ofc Chief of Stf, Studies Ofc
1 DCSPER, ATTN: CPS/OCP
1 The Army Lib, Pentagon, ATTN: RSB Chief
1 The Army Lib, Pentagon, ATTN: ANRAL
1 Ofc, Asst Sect of the Army (R&D)
1 Tech Support Ofc, OJCS
1 USASA, Arlington, ATTN: IARD-T
1 USA Rsch Ofc, Durham, ATTN: Life Sciences Dir
2 USARIEM, Natick, ATTN: SGRD-UE CA
1 USATC, Ft Clayton, ATTN: STTC-MO A
1 USAIMA, Ft Bragg, ATTN: ATSU-CTD-OM
1 USAIMA, Ft Bragg, ATTN: Marquat Lib
1 US WAC Ctr & Sch, Ft McClellan, ATTN: Lib
1 US WAC Ctr & Sch, Ft McClellan, ATTN: Tnq Dir
1 USA Quartermaster Sch, Ft Lee, ATTN: ATSM-TE
1 Intelligence Material Dev Ofc, EWL, Ft Holabird
1 USA SE Signal Sch, Ft Gordon, ATTN: ATSO EA
1 USA Chaplain Ctr & Sch, Ft Hamilton, ATTN: ATSC-TE-RD
1 USATSC, Ft Eustis, ATTN: Educ Advisor
1 USA War College, Carlisle Barracks, ATTN: Lib
2 WRAIR, Neurosychiatry Div
1 DLI, SDA, Monterey
1 USA Concept Anal Agcy, Bethesda, ATTN: MOCA-MR
1 USA Concept Anal Agcy, Bethesda, ATTN: MOCA-JF
1 USA Arctic Test Ctr, APO Seattle, ATTN: STEAC-PL-MI
1 USA Arctic Test Ctr, APO Seattle, ATTN: AMSTE-PL-TS
1 USA Armament Cmd, Redstone Arsenal, ATTN: ATSK-TEM
1 USA Armament Cmd, Rock Island, ATTN: AMSAR-TDC
1 FAA-NAFEC, Atlantic City, ATTN: Library
1 FAA NAFEC, Atlantic City, ATTN: Human Engr Br
1 FAA Aeronautical Ctr, Oklahoma City, ATTN: AAC-44D
2 USA Fld Arty Sch, Ft Sill, ATTN: Library
1 USA Armor Sch, Ft Knox, ATTN: Library
1 USA Armor Sch, Ft Knox, ATTN: ATSB-DI-F
1 USA Armor Sch, Ft Knox, ATTN: ATSB DT TP
1 USA Armor Sch, Ft Knox, ATTN: ATSB-CD AD
2 HQUSACDEC, Ft Ord, ATTN: Library
1 HQUSACDEC, Ft Ord, ATTN: ATEC-EX-E - Hum Factors
2 USAEEC, Ft Benjamin Harrison, ATTN: Library
1 USAPACDC, Ft Benjamin Harrison, ATTN: ATCP-HR
1 USA Comm-Elect Sch, Ft Monmouth, ATTN: ATSN-EA
1 USAEC, Ft Monmouth, ATTN: AMSEL-CT HDP
1 USAEC, Ft Monmouth, ATTN: AMSEL-PA P
1 USAEC, Ft Monmouth, ATTN: AMSEL-SI-CB
1 USAEC, Ft Monmouth, ATTN: C, Fac Dev Br
1 USA Materials Sys Anal Agcy, Aberdeen, ATTN: AMXSY-P
1 Edgewood Arsenal, Aberdeen, ATTN: SAREA BL H
1 USA Ord Ctr & Sch, Aberdeen, ATTN: ATSL-TEM-C
2 USA Hum Engr Lab, Aberdeen, ATTN: Library/Dir
1 USA Combat Arms Tng Bd, Ft Benning, ATTN: Ad Supervisor
1 USA Infantry Hum Rsch Unit, Ft Benning, ATTN: Chief
1 USA Infantry Bd, Ft Benning, ATTN: STEBC-TE-T
1 USASMA, Ft Bliss, ATTN: ATSS-LRC
1 USA Air Def Sch, Ft Bliss, ATTN: ATSA CTD ME
1 USA Air Def Sch, Ft Bliss, ATTN: Tech Lib
1 USA Air Def Bd, Ft Bliss, ATTN: FILES
1 USA Air Def Bd, Ft Bliss, ATTN: STEBD-PO
1 USA Cmd & General Stf College, Ft Leavenworth, ATTN: Lib
1 USA Cmd & General Stf College, Ft Leavenworth, ATTN: ATSW-SE-L
1 USA Cmd & General Stf College, Ft Leavenworth, ATTN: Ed Advisor
1 USA Combined Arms Cmbt Dev Act, Ft Leavenworth, ATTN: DepCdr
1 USA Combined Arms Cmbt Dev Act, Ft Leavenworth, ATTN: CCS
1 USA Combined Arms Cmbt Dev Act, Ft Leavenworth, ATTN: ATCASA
1 USA Combined Arms Cmbt Dev Act, Ft Leavenworth, ATTN: ATCACO-E
1 USA Combined Arms Cmbt Dev Act, Ft Leavenworth, ATTN: ATCACO-CI
1 USAFECom, Night Vision Lab, Ft Belvoir, ATTN: AMSEL-NV-SD
3 USA Computer Sys Cmd, Ft Belvoir, ATTN: Tech Library
1 USAMERDC, Ft Belvoir, ATTN: STSF-B-DO
1 USA Eng Sch, Ft Belvoir, ATTN: Library
1 USA Topographic Lab, Ft Belvoir, ATTN: ETL TD-S
1 USA Topographic Lab, Ft Belvoir, ATTN: STINFO Center
1 USA Topographic Lab, Ft Belvoir, ATTN: ETL GSL
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: CTD MS
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: ATS-CTD-MS
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: ATSI-TE
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: ATSI-TEX-GS
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: ATSI-CTS-OR
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: ATSI-CTD-DT
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: ATSI-CTD-CS
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: DAS/SDR
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: ATSI-TEM
1 USA Intelligence Ctr & Sch, Ft Huachuca, ATTN: Library
1 CDR, HQ Ft Huachuca, ATTN: Tech Ref Div
2 CDR, USA Electronic Prvg Grd, ATTN: STEPP MT-S
1 HQ, TCATA, ATTN: Tech Library
1 HQ, TCATA, ATTN: ATCAT-OP-O, Ft Hood
1 USA Recruiting Cmd, Ft Sheridan, ATTN: USARCPM-P
1 Senior Army Adv., USAFAGOD/TAC, Elgin AF Aux Fld No. 9
1 HQ, USARPAC, DCSPER, APO SF 96558, ATTN: GPPE-SE
1 Stimson Lib, Academy of Health Sciences, Ft Sam Houston
1 Marine Corps Inst., ATTN: Dean-MCI
1 HQ, USMC, Commandant, ATTN: Code MTMT
1 HQ, USMC, Commandant, ATTN: Code MPI-20-28
2 USCG Academy, New London, ATTN: Admission
2 USCG Academy, New London, ATTN: Library
1 USCG Training Ctr, NY, ATTN: CO
1 USCG Training Ctr, NY, ATTN: Educ Svc Ofc
1 USCG, Psychol Res Br, DC, ATTN: GP 1/62
1 HQ Mid-Range Br, MC Det, Quantico, ATTN: P&S Div

1 US Marine Corps Liaison Ofc, AMC, Alexandria, ATTN: AMCGS-T
1 USATRADOC, Ft Monroe, ATTN: ATRO-ED
6 USATRADOC, Ft Monroe, ATTN: APR- AD
1 USATRADOC, Ft Monroe, ATTN: ATTS-EA
1 USA Forces Cmd, Ft McPherson, ATTN: Library
2 USA Aviation Test Bd, Ft Rucker, ATTN: STEBG-PO
1 USA Agcy for Aviation Safety, Ft Rucker, ATTN: Library
1 USA Agcy for Aviation Safety, Ft Rucker, ATTN: Educ Advisor
1 USA Aviation Sch, Ft Rucker, ATTN: PO Drawer O
1 HQUSA Aviation Sys Cmd, St Louis, ATTN: AMSAV-ZDR
2 USA Aviation Sys Test Act, Edwards AFB, ATTN: SAVTE-T
1 USA Air Def Sch, Ft Bliss, ATTN: ATSA TEM
1 USA Air Mobility Rsch & Dev Lab, Moffett Fld, ATTN: SAVDL -AS
1 USA Aviation Sch, Res Tng Mgt, Ft Rucker, ATTN: ATST-T-RTM
1 USA Aviation Sch, CO, Ft Rucker, ATTN: ATST-D-A
1 HQ, DARCOM, Alexandria, ATTN: AMXCD-TL
1 HQ, DARCOM, Alexandria, ATTN: CDR
1 US Military Academy, West Point, ATTN: Serials Unit
1 US Military Academy, West Point, ATTN: Ofc of Milt Ldrshp
1 US Military Academy, West Point, ATTN: MAOR
1 USA Standardization Gp, UK, FPO NY, ATTN: MASE-GC
1 Ofc of Naval Rsch, Arlington, ATTN: Code 452
3 Ofc of Naval Rsch, Arlington, ATTN: Code 458
1 Ofc of Naval Rsch, Arlington, ATTN: Code 450
1 Ofc of Naval Rsch, Arlington, ATTN: Code 441
1 Naval Aerospic Med Res Lab, Pensacola, ATTN: Acous Sch Div
1 Naval Aerospic Med Res Lab, Pensacola, ATTN: Code L51
1 Naval Aerospic Med Res Lab, Pensacola, ATTN: Code L5
1 Chief of NavPers, ATTN: Pers-OR
1 NAVAIRSTA, Norfolk, ATTN: Safety Ctr
1 Nav Oceanographic, DC, ATTN: Code 8251, Charts & Tech
1 Center of Naval Anal, ATTN: Doc Ctr
1 NavAirSysCom, ATTN: AIR-5313C
1 Nav BuMed, ATTN: 713
1 NavHelicopterSubSqua 2, FPO SF 98801
1 AFHRL (FT) Williams AFB
1 AFHRL (TT) Lowry AFB
1 AFHRL (AS) WPAFB, OH
2 AFHRL (DOJZ) Brooks AFB
1 AFHRL (DOJN) Lackland AFB
1 HQUSA (INYSD)
1 HQUSA (DPXXA)
1 AFVTG (RDI) Randolph AFB
3 AMRL (HE) WPAFB, OH
2 AF Inst of Tech, WPAFB, OH, ATTN: ENE/SL
1 ATC (XPTDI) Randolph AFB
1 USAF AeroMed Lib, Brooks AFB (SUL-4), ATTN: DOC SEC
1 AFOSR (INL), Arlington
1 AF Log Cmd, McClellan AFB, ATTN: ALC/DPCRB
1 Air Force Academy, CO, ATTN: Dept of Bel Scn
5 NavPers & Dev Ctr, San Diego
2 Navy Med Neuropsychiatric Rsch Unit, San Diego
1 Nav Electronic Lab, San Diego, ATTN: Res Lab
1 Nav TrngCen, San Diego, ATTN: Code 9000-Lib
1 NavPostGraSch, Monterey, ATTN: Code 55Aa
1 NavPostGraSch, Monterey, ATTN: Code 2124
1 NavTrngEquipCtr, Orlando, ATTN: Tech Lib
1 US Dept of Labor, DC, ATTN: Manpower Admin
1 US Dept of Justice, DC, ATTN: Drug Enforce Admin
1 Nat Bur of Standards, DC, ATTN: Computer Info Section
1 Nat Clearing House for MH- Info, Rockville
1 Denver Federal Ctr, Lakewood, ATTN: BLM
12 Defense Documentation Center
4 Dir Psych, Army Hq, Russell Ofcs, Canberra
1 Scientific Advisr, Mil Bd, Army Hq, Russell Ofcs, Canberra
1 Mil and Air Attaché, Austrian Embassy
1 Centre de Recherche Des Facteurs, Humaine de la Defense Nationale, Brussels
2 Canadian Joint Staff Washington
1 C/Air Staff, Royal Canadian AF, ATTN: Pers Std Anal Br
3 Chief, Canadian Def Rsch Staff, ATTN: C/CRDS(W)
4 British Def Staff, British Embassy, Washington

1 Def & Civil Inst of Enviro Medicine, Canada
1 AIR CRESS, Kensington, ATTN: Info Sys Br
1 Militærpsykologisk Tjeneste, Copenhagen
1 Military Attaché, French Embassy, ATTN: Doc Sec
1 Medecin Chef, C.E.R.P.A.-Arsenal, Toulon/Naval France
1 Prin Scientific Off, Appl Hum Engg Rsch Div, Ministry of Defense, New Delhi
1 Pers Rsch Ofc Library, AKA, Israel Defense Forces
1 Ministerie van Defensie, DOOP/KL Afd Sociaal Psychologische Zaken, The Hague, Netherlands