

NISTIR 7589

**Framework for XML Schema Naming and Design Rules
Development Tools**

KC Morris
Josh Lubell
Salifou Sidi Malick

**Framework for XML Schema Naming and Design Rules
Development Tools**

KC Morris
Josh Lubell
Salifou Sidi Malick
*Manufacturing Systems Integration Division
Manufacturing Engineering Laboratory*

July 2009



U.S. Department of Commerce
Gary Locke, Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Deputy Director

Abstract

Manufacturing industry is facing an enormous challenge today in trying to integrate a wide range of software systems. These systems span the functional areas of manufacturing, as well as, the multitude of organizations and countries involved in the manufacture of their products. To address these needs (and similar needs in other industries) a number of organizations are defining standards that support the integration of the systems. Many of these standards are an application of the XML family of software standards. To ensure quality in the emerging domain standards, the organizations producing them also define guidelines for how the XML standards, especially XML Schema, will be used in a given context. These guidelines are published as sets of Naming and Design Rules (NDR). Unfortunately, a single set of these rules do not meet the needs of the wide range of standards being developed. Moreover, the guidelines are being published as prose (e.g., English text) rather than in a computational form.

To address these shortcomings and allow for the more rapid creation of high quality NDR, NIST is developing a set of tools to facilitate Naming and Design Rules Development. The tools support the development of executable tests to support NDR, collaboration on the use of those tests, the grouping of tests into testing profiles, as well as, the association of NDR documentation with executable tests. The grouping mechanism allows rules to be selected from a larger set for use in a particular context. By also providing for the association of guidelines with executable tests, a testing suite can be quickly assembled. Furthermore, the pool of guidelines and their tests can be collaboratively developed, thereby leveraging resources and creating higher quality test sets in the end.

The report describes a roadmap for the use of these tools in the development of XML Schema Naming and Design Rules. It is produced by the XML Testbed project within the Manufacturing Systems Integration Division at NIST. These tools aid in producing high quality XML schemas using a standards-based approach to manufacturing systems integration. The tools are based on open-standards, making them highly configurable and reusable. The three primary functional areas of the tools described are authoring, testing, and sharing.

Table of Contents

1. Background.....	2
2. Overview.....	3
3. Authoring.....	4
4. Testing.....	5
4.1 Stand-alone Testing	6
4.2 Collaborative Testing.....	6
4.2.1 User and COI	7
4.2.2 Test Requirement	8
4.2.3 Test Case.....	8
4.2.4 Test Profile.....	9
5. Sharing	9
6. Conclusion and Future Work	10
7. Disclaimer	11
8. References.....	12

1. Background

Naming and Design Rules (NDR) are the rules that are applied when creating standards for the exchange of information using XML technology, specifically XML Schema [1]. The rules help ensure that the standards will be clear, consistent, and reusable. Like the standards themselves the rules need to be formulated in the context of business requirements. The rules will change as the context changes. While the XML technology is designed to support a wide range of applications, the rules provide a protocol for use of the technology within a given business context.

As a result of this variability in requirements, a number of different communities have developed distinct NDR documents. The NDR documents come from a wide variety of sources and address the business needs of a given context. Sources for the documents range from formal international standards (such as the United Nations Centre for Trade Facilitation and Electronic Business-- UN/CEFACT[2]) to industrial consortia (such as the Automobile Industry Action Group-- [AIAG](#) [3]), companies, government agencies, and presumably, although not publicly published, internal projects.

Often there is a relationship between NDR documents. For instance, a number of groups have sought to reuse the rules published in the international standards emerging from UN/CEFACT and tailor them for their context [4, 5].

In designing the XML Schema Naming and Design Rules Development Environment several scenarios for NDR documents and their use have been considered. The paper, [XML Schema Design Quality Test Requirements](#) [6], gives a detailed review of the first set of NDR documents that were studied. Since that publication several more NDR documents and their use have been evaluated and tests for the NDR have been developed. In some cases, only a small set of tests have been developed; others have been more

extensive. The more recent NDR documents that have been considered have contributed to the reuse scenarios defined later in this document. These documents include

- [*OAGIS 9 Naming and Design Rules Standard*](#)¹ [4]
- [*Universal Business Language \(UBL\) Naming and Design Rules*](#) [7]
- [*Department of the Navy: XML Naming and Design Rules*](#) [5]
- [*UN/CEFACT XML Naming and Design Rule: Version 2.0*](#) [8]
- [*National Information Exchange Model Naming and Design Rules: Version 1.3*](#) [9]

Given how valuable these rules are, the next logical step is to test whether the XML Schemas that these rules are intended to govern comply with the rules. However, the NDR reviewed do not contain tests for the rules. NIST has been investigating how the rules might be tested and how the tests can be made available in such a way as to serve multiple communities of interest. The experience that NIST has gained in this area has served to improve the quality of forthcoming NDR documents, as well as, provided a basis for creating reusable rules and tests.

The experience has also indicated that the need for tests for NDR is greater than the need for documents of NDR. In other words, while there are several efforts to thoroughly document NDR in formal contexts such as standards organizations, there are many more projects which would benefit from the ability to selectively choose from existing NDR and create a set of rules for their own contexts without the overhead of producing a formal and non-executable document.

To this end this document provides a roadmap for NIST tools to support the use of NDR in developing XML vocabularies. The roadmap explores the expansion of work in testing NDR such that

- creating new sets of NDR based on existing work is facilitated,
- tests of the rules can be reused, and
- documentation can be easily generated.

2. Overview

This roadmap covers 3 functional areas of Naming and Design Rule development:

- Authoring
- Testing
- Sharing

The Quality of Design (QOD) tool is the backbone of the system[10]. The functionality of the first generation of the QOD tool was restricted to the testing area[11]; this roadmap shows how other tools build on the QOD tool to address all three functional areas. In the next generation of the QOD tool (QOD2), emphasis will be given to both testing and sharing. Interfaces to third-party tools will be provided to support the NDR authoring

¹ Open Applications Group Integration Specification (OAGIS)

process. A design principal for the system is to use open standards as much as possible so as to leverage external tools that support those standards.

The authoring component of the system addresses integrating NDR authoring capabilities with a testing environment. Executable tests can be managed in conjunction with the NDR documentation. It also includes mechanisms for rendering NDR documentation in various formats, such as spreadsheets, PDF files, and editable markup using standard Docbook XML [12]. In addition, the authoring capability will include features to support the generation of NDR documentation from rules available in the database.

The testing features relate to the encoding of the rules from an NDR in such a way that an XML schema can be tested for compliance to those rules. Much of this functionality has been covered in the initial version of the QOD tool. A new version expands on this with more features to support authoring and sharing as will be described here. In QOD, NDR rules can be encoded using Schematron [13] or the Java Expert System Shell (JESS) [14]. In the future, support for others of the emerging standards from the World Wide Web Consortium (W3C), such as the XML Query Language (XQuery) [15], may also be included as needed.

The sharing aspects of the system address to how rules can be reused by multiple groups and changed in a manner that does not invalidate existing work using the rules. A number of activities are taking the NDR produced from broad organizations such as UN/CEFACT or the NIEM and attempting to reuse those pieces of work by changing them in a number of ways, either by expansion or restriction, to meet their more detailed requirements. The benefit of leveraging existing (and quite extensive) work is diminished when the base work changes. Any dependent project will need to review all the changes and evaluate how the changes impact them. The changes could invalidate existing schemas. Thus a ripple effect can be seen. The QOD2 tool provides support to manage this type of sharing.

A technical cornerstone for tying these functional areas together is the use of an XML schema for capturing a set of rules, known as the NDRProfile [16]. The NDRProfile captures information necessary to support all three functional areas. It provides a platform for documenting the NDR using generic XML authoring tools in combination with a set of rendering mechanisms for publishing the final result. It also is the format used to exchange rules between various independent components of the system.

3. Authoring

The authoring environment is discussed in detail in *NDRProfile Schema Version 1.0 User Guide* [16]. A short summary is provided here. Authoring capabilities are subdivided into three functional areas:

- Representation
- Presentation
- Editing

Representation addresses how the information is captured in a structured way so that it can be more easily processed by a software application. Presentation refers to various ways in which information can be displayed to an end user. Editing addresses the process of changing the underlying information. In this environment, since the NDR itself is marked up using XML, much of the authoring capabilities rely on the generic capabilities of XML technology.

The foundation of the NDR authoring capabilities is the NDRProfile schema. The NDRProfile schema captures the information needed to support testing and sharing in a structured way such that it can be presented in multiple formats and integrated with a broader set of documentation. A proof of concept demonstration of the authoring capabilities was presented in December 2006 [17].

Several of the presentation tools developed for the authoring demo have been made available for others who would like to create tailored presentations of an NDR. These include Extensible Stylesheet Language (XSL) Transformations[18] (XSLT scripts) for rendering the NDR in various formats including spreadsheets, Portable Document Format (PDF) [19] files, and editable markup using DocBook. See the *NDR Profile Schema Version 1.0 User Guide* for more details.

Editing of an NDR document has typically been done in a traditional word-processing system. The document then needs to be tagged to work with the NDRProfile schema. However, by using an XML native editor with the NDRProfile schema this tagging step can be by-passed.

NDR developers have the ability to author their NDRProfile instance in any tool appropriate to their environment. NDRProfile instances created using a native XML validation tool can be imported directly into the QOD tool. Authors may also create their NDRProfile instance using non-XML software. Other templates (current and proposed) for standard software, such as Microsoft Word and Excel, along with transformation scripts (XSLT) will give the author the ability to convert non-XML NDR to NDRProfile tagged XML files for importing into the QOD tool.

4. Testing

The two primary use cases for testing considered in the design of the QOD2 tool are characterized as stand-alone and collaborative.

In the stand-alone use case a schema developer wishes to test their schemas against a single set of rules and is not interested in sharing the results of the tests. This scenario may be supported by a stand-alone system that does not depend on connections to the Internet.

The collaborative use case can be further subdivided into support for test developers and schema developers. Test developers may use the system to coordinate with both other test developers and with schema developers. The system can be used to have other test developers validate new tests before the tests are deployed to the schema developers. Furthermore, new tests may be added to test sets as the schemas are being developed, providing schema developers access to the latest version and set of tests. Finally, schema developers may wish to share their test results with others working on the schema. All of these scenarios are supported with Internet access to a database of tests, test sets, and test results.

4.1 Stand-alone Testing

Stand-alone testing is done by using an instance of an NDRProfile. The NDRProfile contains both the text of the NDR document as well as test scripts. The document can be rendered in multiple formats for viewing as described in the authoring section, and the test scripts can be extracted for execution against an XML schema. The test scripts can be executed on a schema using third-party tools since the scripts are written using well supported languages. NDRProfiles can be exported from the QOD tool for use in stand-alone testing. NIST also has a stand-alone testing system which runs on a personal computer [20].

It should be noted that the stand-alone use case may also be supported by the more elaborate system described below for collaborative testing; however, the full system is not necessary.

4.2 Collaborative Testing

The QOD tool is designed to support the collaborative use case. The QOD tool provides a user interface accessible over the Internet and a shared database. Access to the database is through the web pages. Figure 1 shows an object model which expands on the original QOD tool to better support multiple users and user groups. As is seen in Figure 1 five concepts are fundamental to the testing aspects of the system:

- User: the person using the system
- Community of Interest (COI): a group of people working together to develop an NDR
- Test Requirement (TR): a statement in ordinary language of the requirement or rule
- Test Case (TC): a computer-encoded form of the requirement
- Test Profile (TP): a collection of test requirements

Other objects in this model represent elaborations of and relationships between these five concepts.

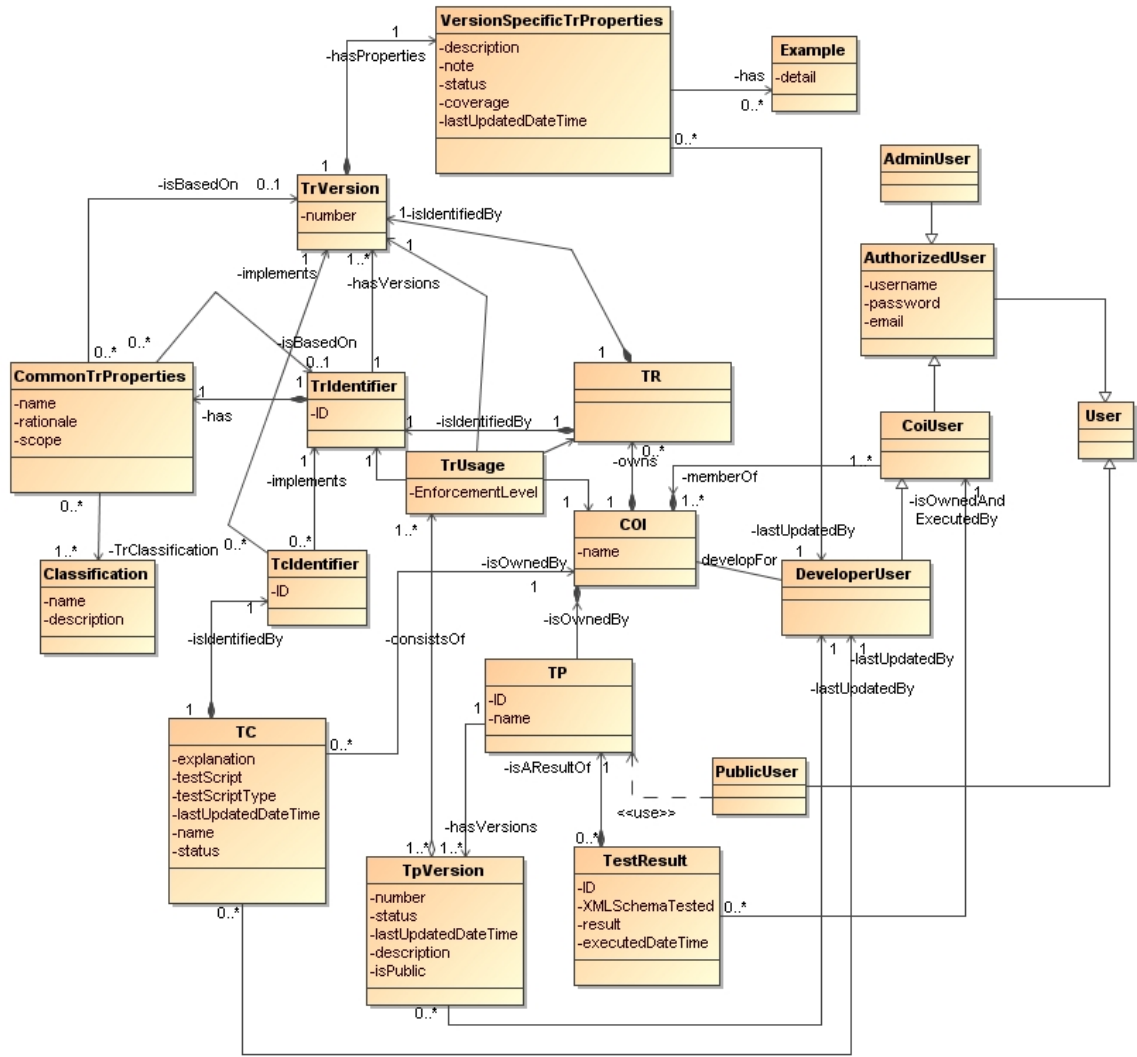


Figure 1: Object Model for QOD2

Two of these concepts, *User* and *COI*, are unique to the collaborative environment. NDR documents can generally provide the content for the Test Requirement attributes (i.e., the textual statements of the rules.) The system extends what is captured in the documents in its capacity to store and associate test cases with the requirements, organizes the requirements into collections, and facilitates collaboration among multiple users.

Each of the latter three concepts—test requirement, test case, and test profile—is discussed in detail in the QOD system documentation, <http://syseng.nist.gov/b2bTestbed/projects/XMLVT/html/xmlvtMaterial.jsp?session=qod-testReq>. The descriptions below refer to how the concepts are expanded in QOD2 tool.

4.2.1 User and COI

In the QOD tool, the idea of the user is primitive. Users create and own test requirements, test profiles, and test cases. The user can opt to make the test profiles and test cases

available to the public or not. Only the owners of these artifacts can modify and delete the items.

In the enhanced version of the QOD tool described in this paper the notion of a user is expanded to include the concepts of a Community of Interest (COI) and developers. While individual users are able to create test requirements and test profiles, these artifacts are owned by the COI's developers. The COI developer group shares the rights of ownership and is able to modify and delete the items owned by the group. This allows for close collaboration without exposing the items to the broader set of users (i.e. public users.) Moreover, other members of the COI (i.e., non-developers) may view the items and execute the tests even when they are not available to the general public.

The tool also supports an administrative user. The administrative user may add new users, COIs, NDR documents and version identifiers to the system.

4.2.2 Test Requirement

A *test requirement* defined as a guidance element in the NDRProfile schema, is a non-executable statement or assertion. Each test requirement may have one or more corresponding test cases; however, not all test requirements are testable (at least not programmatically testable), so there may be test requirements without test cases. For example, an untestable test requirement is that “an attribute with variable information must be based on the appropriate built in XML Schema data type.” We have not determined a good way to test for “appropriateness.” Test Requirements may be visible to all users (i.e., including public users) or the visibility may be limited.

Test requirements are typically based on specific best practices drawn from experienced system integrators and/or XML architects and are often published in NDR documents. The requirements seek to improve the usability, re-usability, and interoperability of the schema by enhancing extensibility, ease of maintenance, implementation and processing efficiency, and the schema's ability to capture and enforce desired constraints.

The QOD2 tool allows for versioning of test requirements and allows test cases to propagate to new versions of a test requirement.

4.2.3 Test Case

A *test case* is a computer executable script (i.e., the binding of a test requirement to a specific implementation.) Generally, a test case may contain multiple test steps. Multiple test cases may be associated with the same test requirement. Reasons for this include different reference data, different scopes, or different rule languages used to implement different aspects of the test requirement. Additionally, when a test requirement is reused, the group reusing the test requirement may wish to add a test case to implement refinements to the test requirement and at the same time reuse the test case from the original test requirement. It is recommended that given a choice a single test case should be used to implement a test requirement. Limiting the number of test cases simplifies management of the test cases. In the QOD tool, when there is a test profile referencing a test requirement and, consequently, a test case, the test case cannot be

deleted. The status field of the test case may be used to indicate that it is no longer recommended.

4.2.4 Test Profile

A *test profile* is a mechanism for grouping related test requirements. For example, one profile might contain requirements for a schema designed to be building blocks for other schemas, while another may check for a schema designed for a specific type of document. In this case, there may be only minor differences between the profiles, such as with rules checking for a final form of the namespace(s)[11]. When the user executes a test profile, all the included test cases are executed.

The QOD2 tool supports versioning of test profiles.

5. Sharing

Three types of NDR are considered in this roadmap:

- Formal: produced by standards associations or other organizations and intended to be the primary source of guidelines for XML schema developers.
- Derived: produced by projects or other communities of interest with the intent of reusing and customizing the rules from formal NDR documents, with or without the intent of producing a full set of independent documentation.
- Test: produced as part of a project by using a subset of rules from one of the other types of NDR.

In the case of the formal NDR, the systematic use of the NDRProfile allows one to coordinate the testing with the rest of the information that makes up an NDR document. Similarly, in the case of the derived and test NDR, the use of the NDRProfile allows for traceability to the original source of the rules. The systematic use of the NDRProfile depends on the rule identifiers.

In the terms of the NDRProfile schema, the rules are referred to as *Guidance*. Every guidance element has an identifier. This identifier is obtained by taking an identifier for the source document-- the formal NDR-- including an indicator for the version of that document, and combining it with a unique identifier within that document. Thus each rule, whether in a reused NDR or a formal NDR, is traceable back to its primary source documentation.

A single source NDR (i.e., the formal NDR) can be presented through multiple profiles which test for a cluster of features, and result in a number of Test NDRs. In this scenario no further documentation of the rules themselves is called for. The identifiers still provide traceability to the original source with the associated documentation.

In all these cases the basic unit of sharing is the test requirement. Test requirements are owned by a COI and may be made public, in which case it is viewable by everyone who

has access to the system. Conversely, the visibility of the test requirement can be limited to the COI.

The use cases supported by the QOD2 tool are illustrated in the Figure 2.

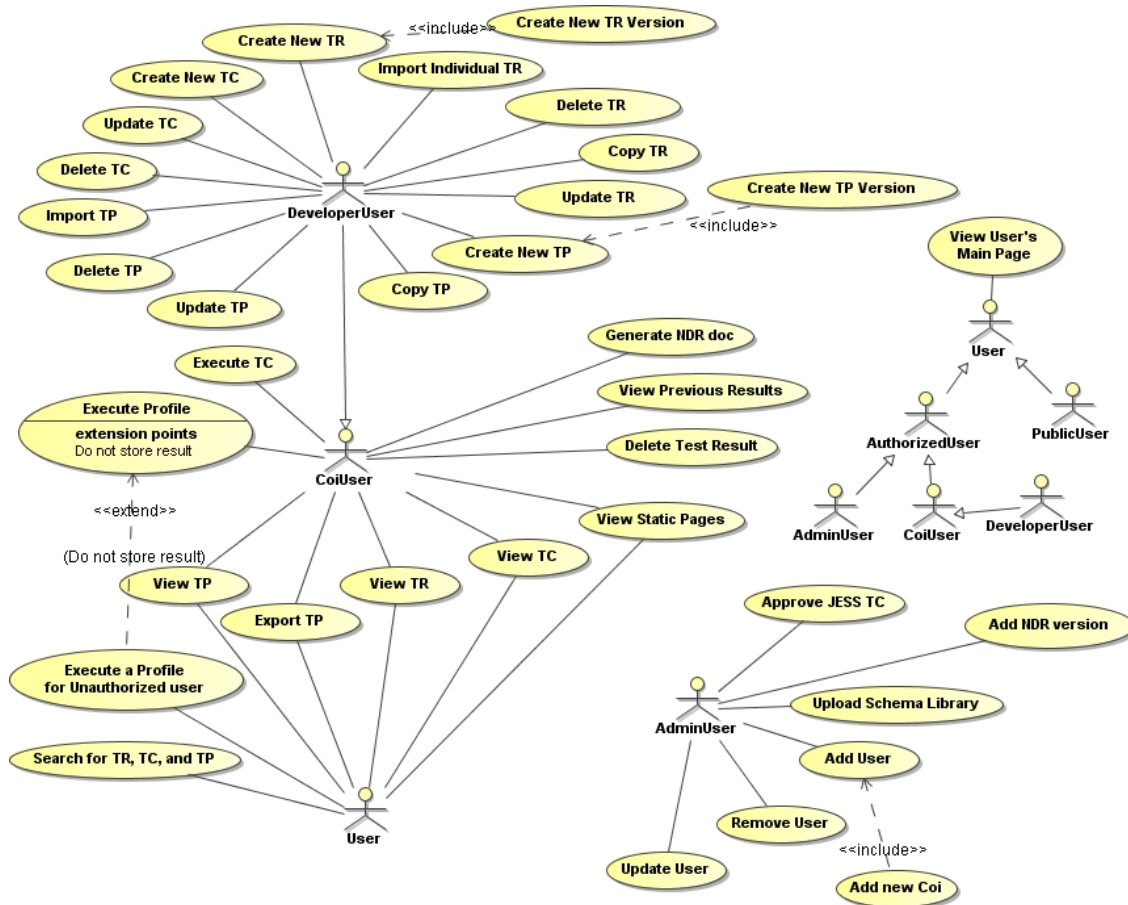


Figure 2: Use Case Model for QOD2

6. Conclusion and Future Work

This framework for tools to support the development of NDR is based on open standards so as to enable different tools to work together and to work with third-party tools, thereby supporting the underlying technologies as much as possible. While there are a number of robust tools in the areas of XML, XML Schema, Schematron, XSLT, PDF, and others, those tools are not designed to serve the specialized task of NDR development and compliance. This framework fills a niche in providing mechanisms to support that functionality and outlining how these other tools can be leveraged.

As the number of groups using this approach to system integration grows, the demand for this type of framework will also grow. Software vendors are motivated to provide tools

based on the underlying technologies because the markets for these technologies are relatively large, but the niche of developing systems integration standards is much smaller. Furthermore, since NDR are tailored to the needs of specific integration projects, a one-size solution that a vendor may provide is not flexible enough to support the breadth of projects covered with this framework. Moreover, the community approach to developing and sharing of NDR allows for the development of rules and tests that may be leveraged, dispersed, and reused. A centralized location for collecting guidelines on the use of XML Schema for standards development and systems integration provides an experience base for future activities to learn from and reuse.

As this framework is put into use and we continue to work with partners to better understand the needs for standards related to systems integration, we will continue to see areas where the work can be expanded on and generalized to address different aspects of the systems integration problem. Some activities are underway to explore future possibilities in the following areas:

- With regard to the QOD database of NDR two enhancements may be included as usage increases: 1) metrics on rule use and reuse will be captured so as to identify what rules are frequently used, and 2) in order to assist in managing the standards landscape, notifications could be set up so that users of NDRs will know when the rules that they are using change.
- In addition to supporting XML, we are exploring whether and how these systems can be expanded to support other messaging formats.
- The design principal upon which the QOD tool is based is easily generalizable into the idea of testing data, actual instances of XML schemas. In the case of the QOD tool, the XML schema on which the testing is based is the XML schema for XML Schema (xmlschema.xsd), but the same concepts can apply to other XML schema. Thus, we have reused the software for a tool called the Content Checker [11]. The use of the NDRProfile concept for this tool is being explored. Initial findings suggest that the concept can be expanded to provide a mechanism for capturing and documenting business rules, i.e. rules on how data would be used in a transaction based on an XML schema for exchange.

7. Disclaimer

Mention of commercial products or services in this paper does not imply approval or endorsement by NIST, nor does it imply that such products or services are necessarily the best available for the purpose.

The work described is funded by the United States Government and is not subject to copyright.

8. References

1. World Wide Web Consortium; *XML Schema 1.0*; [cited 2008]; available from <http://www.w3.org/XML/Schema>.
2. United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) Techniques and Methodologies Group Home Page; [cited 2005]; available from <http://www.ifs.univie.ac.at/untmg/>.
3. Automobile Industry Action Group Home Page; [cited March 2009]; available from <http://www.aiag.org/>.
4. Open Application Group; *OAGIS 9.0 Naming and Design Rules*; available from <http://www.openapplications.org/>.
5. Chief Information Officer; *Department of the Navy XML Naming and Design Rules: Final Version 2.0*; [cited August 2008]; available from <http://xml.coverpages.org/DON-XML-NDR20050127-33942.pdf>.
6. B. Kulvatunyou and K. Morris; *XML Schema Design Quality Test Requirements*; NISTIR 7175; National Institute of Standards and Technology; Gaithersburg, MD; October 2004; available from http://www.mel.nist.gov/msidlibrary/doc/Schema_Design.pdf.
7. *Universal Business Language (UBL) Naming and Design Rules*; prd-UBL-NDR-2.0; Organization for the Advancement of Structured Information Standards (OASIS); September 8 2006; available from <http://docs.oasis-open.org/ubl/prd-UBL-NDR-2.0.htm>.
8. United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) Applied Techniques Group (ATG); *XML Naming and Design Rules -- Version 2.0*; United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT); 2008; available from <http://xml.coverpages.org/UN-CEFACT-XML-Naming-and-Design-Rules-V20.pdf>.
9. NIEM Technical Architecture Committee; *National Information Exchange Model Naming and Design Rules*; Version 1.3; National Information Exchange Model October 31, 2008; available from <http://www.niem.gov/pdf/NIEM-NDR-1-3.pdf>.
10. *MSID XML Testbed Home Page*; [cited September 3, 2008]; available from http://www.mel.nist.gov/msid/XML_testbed/index.html.
11. K. Morris, et al.; *User's Guide for the Quality of Design Testing Tool and the Content Checker*; NISTIR 7538; National Institute of Standards and Technology; Gaithersburg, MD; November 2008; available from http://www.mel.nist.gov/publications/view_pub.cgi?pub_id=824715.
12. N. Walsh; *DocBook.org*; [cited 2009]; available from <http://www.docbook.org/>.
13. R. Jelliffe; *Schematron*; last updated August 7, 2008; [cited 2008]; available from <http://www.schematron.com/>.
14. E.J. Friedman-Hill; *Jess, the Java Expert System Shell*; Version 5.1; Sandia National Laboratories; Livermore, CA; April 24 2000; available from http://web.njit.edu/all_topics/Prog_Lang_Docs/html/jess/.
15. World Wide Web Consortium; *XQuery 1.0: An XML Query Language*; last updated 23 January 2007; [cited 2009]; available from <http://www.w3.org/TR/xquery/>.
16. J. Lubell, et al.; *NDRProfile Schema Version 1.0 User Guide*; NISTIR 7547; National Institute of Standards and Technology; Gaithersburg, MD; December 16,

- 2008; available from
http://www.mel.nist.gov/publications/view_pub.cgi?pub_id=824742.
17. K. Morris; *XML Schema Interoperability Working Group Status Report 12-14-2006*; last updated December 14, 2006; [cited April 2009]; available from <https://collab.core.gov/CommunityBrowser.aspx?id=17542&lang=en-US>.
 18. World Wide Web Consortium; *Extensible Stylesheet Language Transformations (XSLT)*; last updated November 16 1999; [cited 2009]; available from <http://www.w3.org/Style/XSL/>.
 19. International Organization for Standardization (ISO); *ISO 32000-1:2008 Document management — Portable document format — Part 1: PDF 1.7*; International Organization for Standardization (ISO); July 2008; available from http://www.adobe.com/devnet/pdf/pdf_reference.html.
 20. J. Cuvillier and K. Morris; *QOD Standalone Users Guide*; NISTIR; National Institute of Standards and Technology; Gaithersburg, MD; forthcoming;