
Development Life Cycle and Tools for XML Content Models

Serm Kulvatunyou <serm@nist.gov>

Katherine Morris <kcm@nist.gov>

Buhwan Jeong <bjeong@nist.gov>

Puja Goyal <pgoyal@nist.gov>

Abstract

Many integration projects today rely on shared semantic models based on standards represented using Extensible Mark up Language (XML) technologies. Shared semantic models typically evolve and require maintenance. In addition, to promote interoperability and reduce integration costs, the shared semantics should be reused as much as possible. Semantic components must be consistent and valid in terms of agreed upon standards and guidelines. In this paper, we describe an activity model for creation, use, and maintenance of a shared semantic model that is coherent and supports efficient enterprise integration. We then use this activity model to frame our research and the development of tools to support those activities. We provide overviews of these tools primarily in the context of the W3C XML Schema. At the present, we focus our work on the W3C XML Schema as the representation of choice, due to its extensive adoption by industry.

Table of Contents

1. Introduction	1
2. Model Development Life Cycle	3
3. Activities of the Model Development Life Cycle	4
3.1. Model Discovery	5
3.2. Model Validation	6
3.3. Model Piloting	8
3.4. Model Registration	9
3.5. Model Integration	10
4. Conclusion and Future Work	11
5. Product Disclaimer	11
6. Best Practices References	11
Bibliography	12

1. Introduction

The research described in this paper is inspired from our interactions with industry partners within the Manufacturing B2B Interoperability Testbed [B2BTestbed] and the Automated Equipment Exchange (AEX) Testbed project [AEX] at the National Institute of Standards and Technology (NIST). Within these testbeds, we have experienced that inform-

ation exchange specifications, particularly for business application integration, evolve as the integration project proceeds.

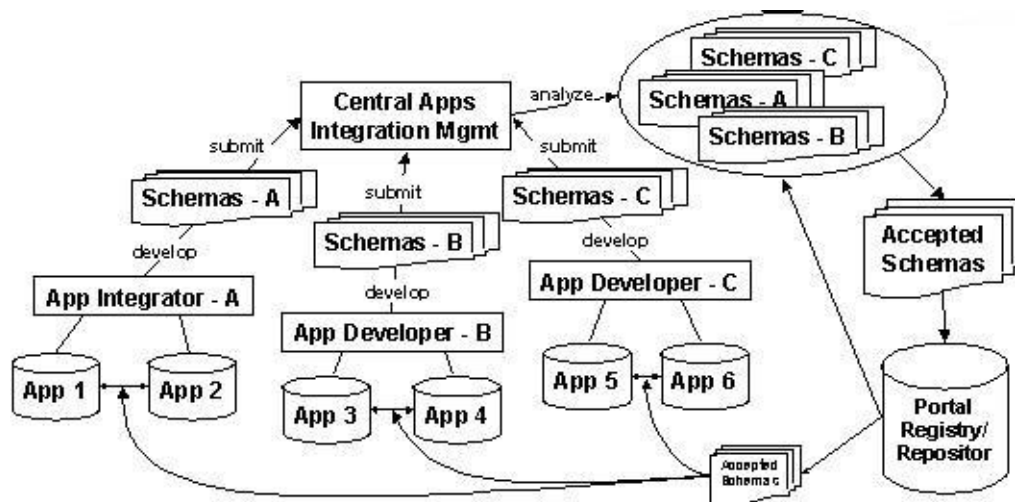


Figure 1. Evolution of information exchange specifications in a large integration project

Figure 1 depicts the evolution of specifications in a large integration project. This figure was generated based on interactions with an organization participating in the B2B Testbed. The enterprise integration involves more than six hundred existing systems and ten of thousands of current point-to-point connections. In such a large enterprise, multiple programs are developed in parallel. A central management team, a registry, and a repository are devised to ensure consistency among schemas that are customized or developed by these programs. In the process to ensure consistency among schemas, each program should try to discover and reuse existing schemas from the repository. If the discovered schemas do not meet requirements, they may be extended or new schemas may be created. The schema extensions or new schemas should then go through tests to ensure that they meet architectural requirements and that they are not duplicating existing schemas. New schemas may be modified or merged with existing ones after careful consideration. They are then registered into the repository for reuse by other programs.

In fact, a similar process occurs throughout content standard development. Taking the Open Application Group (OAG) consortium [OAG] as an example, its business document specification [OAGIS] is constantly extended by its industrial participants. In one case, users in the automotive retail (dealers) sector extended the OAG purchase order standard for replacement part order [STAR2003]. In another case, users in an automotive supply chain extended the OAG order item for use in an inventory management application [Ivezic et. al 2004]. In such cases, the OAG is the central authority that constantly merges new extensions into its core semantic model for each of its subsequent releases.

Current efforts are also underway to harmonize standards for data exchange of business information at the national and international levels. At the national level, the Federal Enterprise Architecture initiative is attempting to centrally control the integration approaches among the US federal agencies [FEA]. At the international level, the UN/CEFACT TBG 17 [TBG17] has been given the authority to harmonize standards to the Core Component Specification [EbCC]. UN/EDIFACT undertook a similar effort for the Electronic Data Interchange (EDI) standard [BSR].

Defining and executing the required activities to achieve this central control has been done on an ad hoc basis. This paper attempts to capture these activities in a well-established process and to represent them using a formal representation, IDEF0 [IDEF0]. The resulting model defines the required activities and their interrelationships. It is used to guide the development of a semantic model that provides the consistency across the activities, and tools to support their execution.

To scope this model, we limited our attention to the activities surrounding systems integration using XML. We also consider authoring of XML Schemas as it relates to integration. Requirements gathering and analysis, which are crucial

to the success of integration, are not included. Before describing our IDEF0 model, we provide the following important notions

- An XML Schema is mainly a syntactic device. It is not capable of representing the entire model's semantics. In order to represent semantics, the schema must be augmented with additional information. This information may take the form of rules, visual models, ontologies (meta-data), supporting documentation, as well as the programming logic.
- XML validation and the processing of an XML instance document are distinct from one another. Furthermore, the processing of instance data can be independent of a given schema language. In other words, an XML instance document can be validated against multiple schemas, perhaps specified using languages other than XML Schema (e.g., Schematron, RELAX NG ISO 2003). In fact, validation can be thought of as a pipeline of various steps, where an application or user processes the XML instance document after it has completed all of its validation steps.
- XML validation refers to the validation of instance data represented in an XML instance document but model validation refers to validating an XML Schema against specification of the XML schema and the requirements of the system or systems to be integrated.

2. Model Development Life Cycle

In this section, we describe the highest-level activity, called the *Model Development Life Cycle*, and its inputs and outputs, which are shown in Figure 2.

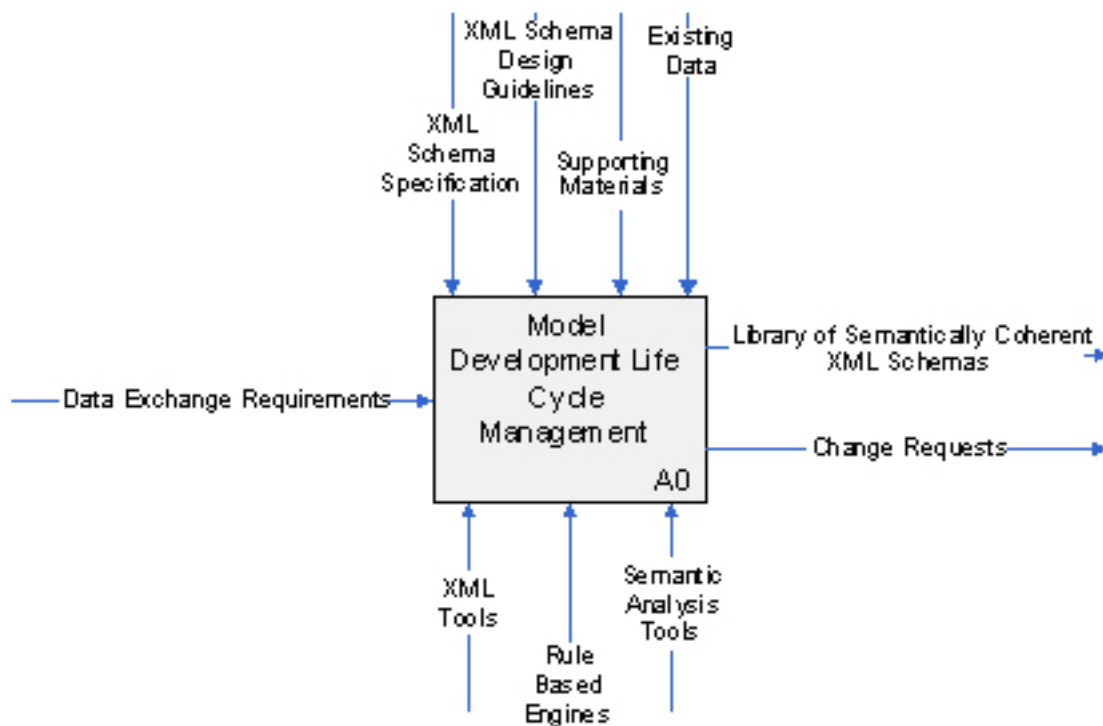


Figure 2. Activity A0 - Model Development Life Cycle

The *Data Exchange Requirements* input includes all the detailed information requirements for integration. This information can be captured in a number of different models: use case models, integration activity models, object/information models, and process models.

The *Library of Semantically Coherent XML Schemas* output is a collection of data interchange terms and data structures represented as XML Schemas [XSD]. Terms and data structures should contain unique semantics or overlapping semantics, but no duplicate semantics. Overlapping can come about by extension, restriction, redefinition, or subsumption. Where direct relationships cannot be established or duplication cannot be eliminated, the models should be formally annotated. The library may incorporate XML-based content standards and will include new XML content models. The resulting library should also contain supporting data to help maximize the reusability of its terms and data structures. Supporting data include but are not limited to, classification schemes for categorization, the models provided in the information exchange requirements, sample XML instance reference data, meta-data, more expressive semantic models, and documentation.

The *Change Requests* output reflects the cyclical nature of a typical life cycle. For example, a request is made to the owning entity to modify their model in order to fully cover requirements or maintain consistency. This results in a change to the existing library.

XML Schema Specification controls the syntactical and grammatical representation of terms and data structures for the data exchange specification. It also limits the expressiveness with which the relationships between overlapping data structures can be modeled.

XML Schema Design Guidelines force compliance of the resulting XML Schemas to a selected set of design principles. These design principles may be preferred ways of utilizing the XML Schema specification when alternatives exist, common data structure patterns, or required meta-data. While some of the guidelines appear to be mere stylistic options, their consistent use is critical to supporting schema reuse. These design guidelines bring a low level of consistency to the resulting schema and facilitate analysis, usability, extensibility, maintainability, automation, and expressiveness.

Supporting material is the collection of source material for understanding the systems, data, and specifications involved in the integration. It may include implementation documentation, business rules, classification schemes, and external domain ontologies that help clarify the intent of the data and specifications.

Although *Existing Data* may be viewed as part of the data exchange requirements input, the purpose here is as reference information to support requirement satisfaction and compatibility analyses.

XML Tools encompass tools that implement the XML Schema specification and related XML standards. These include XML schema validators; XML parsers and validators, and editors; and, other tools that implement utility standards related to XML such as the XML Path language [XPath] and the Extensible Stylesheet Transformation Language [XSLT].

Rule-based Engines are mechanisms used to test conformance of schemas to design guidelines and other requirements. Schematron [Jelliffe 2003] is a specific example of a rule-based engine that is used with XML standards including the XML Schema. A traditional rule-based expert system such as the Java-based Expert System Shell [Friedman-Hill 2002] may be used as well.

Semantic Analysis Tools enhance reuse of the semantic model or XML Schemas. They may support discovery, harmonization, and library management and maintenance.

3. Activities of the Model Development Life Cycle

The Model Development Life Cycle Activity *A0* is decomposed into the five subactivities shown in Figure 3. These activities, *A1 - Model Discovery*, *A2 - Model Validation*, *A3 - Model Piloting*, *A4 - Model Registration*, and *A5 - Model Integration*, are described in subsequent sections.

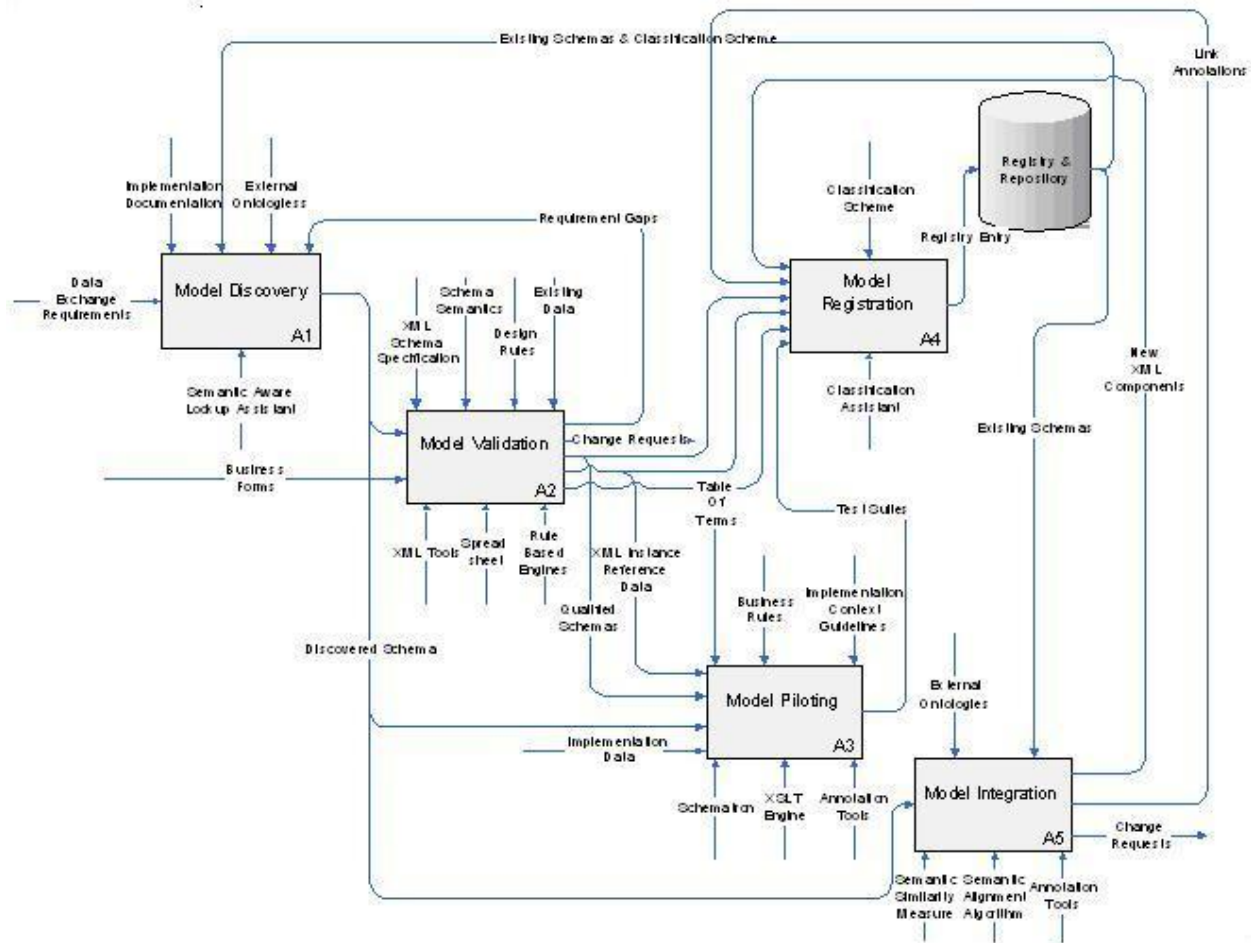


Figure 3. Decomposition of the Model Development Life Cycle

3.1. Model Discovery

Typically, integration projects first try to identify relevant, existing XML Schemas. This process is called *Model Discovery* and is depicted in Figure 4. The initial activity is *Model Selection*. This is followed either by *Model Extension*, when a suitable model has been found, or *Model Creation*, when no appropriate model is available.

Model Selection, which involves finding a relevant, pre-tested, existing model, can be difficult. Consequently, integration engineers often skip this activity and go directly to *Model Creation*. Nevertheless, the first activity should always be to discover any model that fits the scope of the project as captured in the data exchange requirements. To facilitate this process, we envision a tool called a *Semantic Aware Lookup Assistant*. This tool would operate on the *Known Schemas* registered in a model repository using one or more classification schemes (see the *Model Registration* activity below). It will provide a search capability that goes beyond keyword search in the schema or the *Schema Documentation*. For instance, it may provide a guided search based on question-and-answer interactions with the user. The questions asked would be based on the artifacts stored in the registry and the contexts used to drive the semantics associated with the schemas. The tool would also use *External Ontologies* related to the schemas to improve the matching.

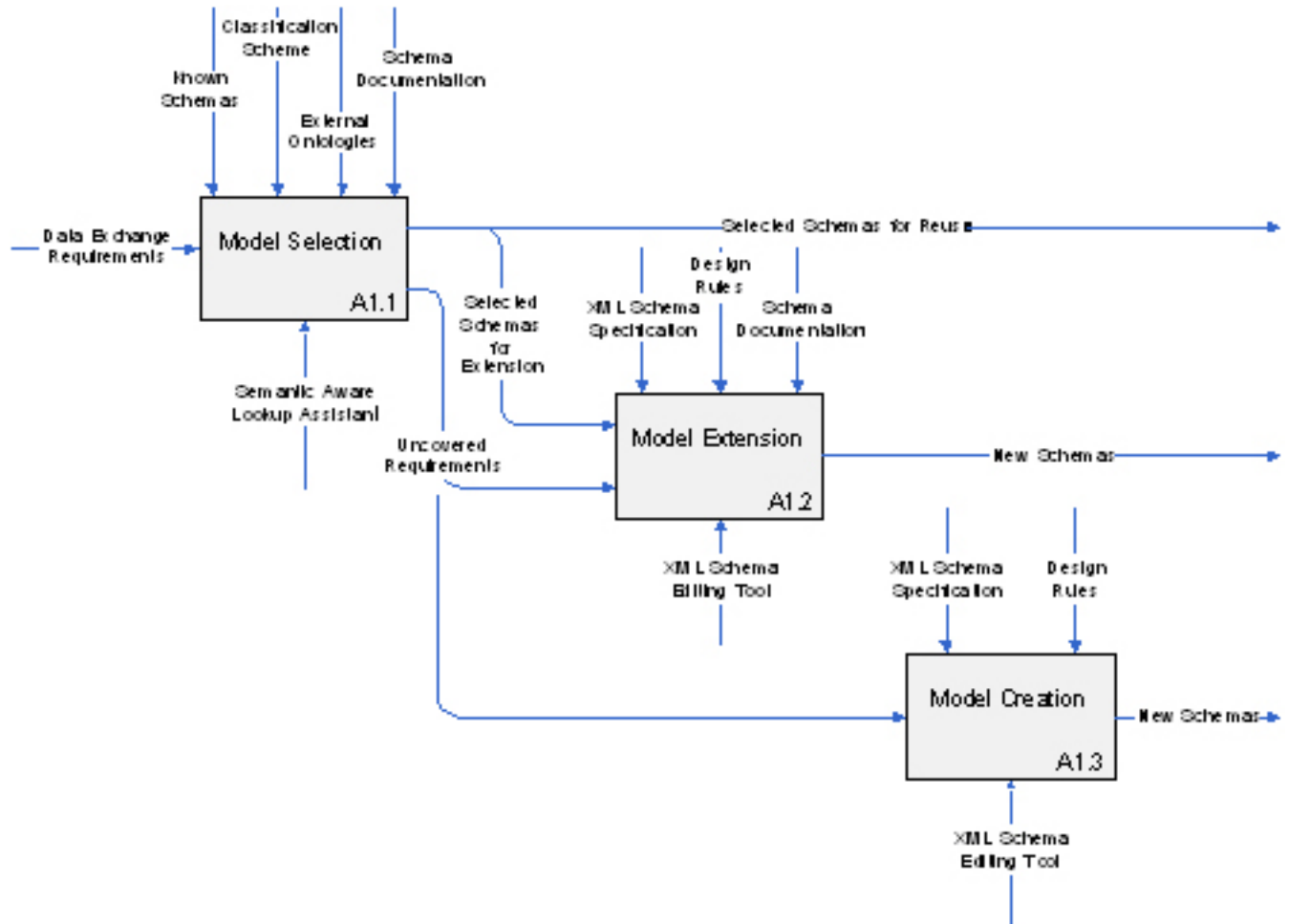


Figure 4. Activity A1 - Model Discovery

When a relevant model has been identified, it may be reused "as is" or it may need some minor modification. This is handled in activity A1.2. The need for extension can be determined by analyzing the extent to which the selected model covers the data exchange requirements for the project. During this activity, implementation documentation will also guide the processes involved in extending the schemas.

Activity A1.3 *Model Creation* is relatively straightforward and can be done using several publicly available tools. Some of these tools may be customized to tightly integrate with the data model and XML Schema design guidelines to assist the schema developer. Both the *Model Creation* and *Model Extension* activities result in new XML Schema files, which should then be validated as described below. The *XML Schema Specification* and *Design Rules* constrain the new schema files created. The *Schema Documentation* of the selected schemas for reuse also guides the model extension as it contains more precise semantics about the schema elements.

3.2. Model Validation

The *Model Validation* activity takes as input an initial information specification, such as the XML schema, produced by the *Model Discovery* activity. Since an XML schema is not like other pieces of software - it has no execution requirements - the *Model Validation* activity includes tests for requirements coverage and design quality (see Figure 5).

Figure 5. Activity A2- Model Validation

Model Validation involves two types of quality validations. The first validation, represented in activity A2.1, is *Schema Qualification*. In this activity an XML Schema is tested against the standard specification for XML Schemas, for example using the XMLSchema.xsd [XSD] through *XML validation tools*. The XML schema is also checked for compliance with the project's design rules and naming conventions using *Rule-based Engine* and the *Naming Assistant*. Naming conventions may be viewed as a form of design guidelines. Modeling guidelines should be established, documented, and enforced as early as possible in the model development in order to avoid rework. These types of compliance checks ensure that modeling practices are used consistently. This enhances the specification's intelligibility and avoids confusion during the pilot and implementation phases of the integration project.

To support quality validation NIST has built prototypes of three proof-of-concept tools: the naming assistant tool, the XML schema quality of design tool, and the XML validation page tool. These tools are described below.

- One result of the schema qualification activity is a table of terms to be used for naming in the XML schema. An initial table may have been provided by the model discovery activity. NIST has built a prototype of a tool, the Naming Assistant, to help with the XML tag naming [Goyal 2004]. More specifically, the tool aids in creating consistent compound names by verifying the construction of these names against a table of allowable terms. The form of the table is based on extensions to the International Organization for Standardization (ISO) -11179's [ISO2004] recommended naming convention and was developed for the AEX Testbed. The tool also assists in establishing a table of standard terms.
- The XML Schema Quality of Design Tool [Kulvatunyou et. al 2004] is a diagnostic tool for checking an XML Schema for compliance with the encoded guidelines. The guidelines are stored in a repository of rules and a framework to publish and execute design rules. The repository has been loaded with an initial set of such rules based on published "Best Practices" [see "Best Practices" References] guidelines for XML authoring. Based on our initial experience in using the tool, we believe that it can be extended to support a larger set of rules, more complex rules, and the capability of creating an extensible rule set, which can be tailored to the requirements for specific projects.
- The final prototype tool built by NIST is the XML Validation page, which allows a user to upload XML instance files and have them validated against the content of a particular set of XML Schema files. Currently, the tool operates over a repository of XML Schema files for a specific project, the NIST AEX Testbed.

Activities A2.2 and A2.3 represent the second type of validation that ensures the model meets the original data exchange requirements. The most direct way of achieving this is to analyze the relationship between an XML schema and the application data. Activity A2.2 gathers existing application data, generates *XML Instance Reference Data*, and identifies gaps between the instance data and the XML schema. This is primarily a manual process that uses a spreadsheet to map from data exchange requirements into the XML schema, and vice versa.

Activity A2.3 deals with the case when the schemas have been over specified with respect to the constraints, meaning that they invalidate the instance data. This typically happens because there are problems in the XML schemas or their supporting material and not in the instance data. For example, the schema may be too restrictive or too ambiguous. Resolution of such problems should result in improvements to either the integration schema or the supporting documentation. The outputs of activity A2.3 can be the *Revised XML Instance Reference Data* that has been slightly adjusted to fit the schemas, schemas that have been improved (*Qualified Schema*), or the *Change Requests* for changes sanctioned by the community. This activity can be viewed as a maintenance step as well. When changes are made to the schema, it is prudent to ensure backward compatibility with the existing (reference) data.

Model Validation (A2) can take much iteration of its sub-activities, but the end result is a valid schema meeting a given set of quality criteria along with documentation describing the schema and how it is to be used. That documentation contains reference data and the table of terms representing the naming conventions. These three outputs provide important inputs to Activity A5 *Model Registration*, which is discussed in section 3.4.

3.3. Model Piloting

To solve a real integration problem, we must exchange information between specific software applications, which may impose additional requirements on the XML schemas. So, while the schemas in our repository presumably cover most of these additional requirements, certain modifications may be necessary. For one XML Schema this may involve additional usage criteria specific to the applications to be integrated. For another, it may also involve a simplification to make it directly applicable to the applications to be integrated. The *Model Piloting* activity deals with these issues by making the available schemas usable for the specific integration problem at hand. Figure 6 illustrates the three subactivities of *Model Piloting*.

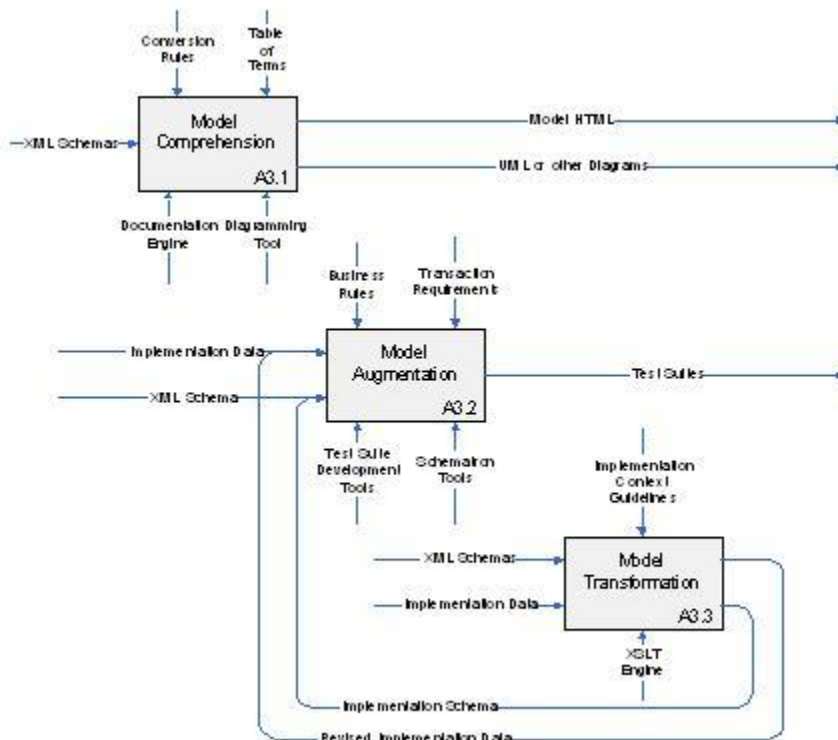


Figure 6. Activity A3 - Model Piloting

Model Comprehension develops an understanding of the schema as it relates to the specific integration problem. Several types of tools, which generate various views of an XML schema based on different *Conversion Rules*, can assist the user. For example, one such tool can create HTML pages that list and connect the various definitions in the schema through hyperlinked text [XSDDOC]. Another tool can produce class diagrams of the structures defined in the schema [HyperModel].

Model Augmentation captures and codifies the requirements that apply to each transaction. They may not be applicable to all industries or to all transactions, yet they may apply at various times and for various purposes. NIST has prototyped two *Schematron Tools* --the Content Checking Tool and the Schematron Editor--to assist in this process in our B2B Testbed [CCT, CSWizard]. The outputs from this activity include a test suite including the implementation schema, instance data, additional rules for validating the data based on the integration context, and guidance on how to use the schema in that context.

Finally, activity A3.3 addresses *Model Transformation*. During *Model Transformation* an XML Schema can be transformed in a systematic way (using the *XSLT Engine*) to support the needs for a particular *Implementation Context*. Examples of when this may be desirable include the following scenarios:

- A project replaces the names used in a standard by terms more common to the businesses involved in the integration.
- An implementation group decides to use a single namespace or a namespace other than the one defined in the standard; this can also be accomplished through a transformation.
- An implementation group may prefer to work with a language other than XML Schema, such as Document Type Definitions (DTDs).

Transformations may be performed on both schema and instance data resulting in a revised schema suitable for a specific implementation and revised reference data that corresponds to that schema. Additionally, data from an outside source, which conforms to the original model, may need to be transformed to fit the modified schema.

The *Model Piloting* activities may or may not result in changes to the original XML schemas. They should result, however, in improved artifacts such as better documentation, better and more robust instance data, and guidelines on how to use the XML Schema in a given business context.

3.4. Model Registration

The *Model Registration* activity organizes the schemas and related materials within a registry and stores them in a repository that is accessible to other activities (see Figure 7). Other supplemental information such as schema version, hierarchical namespace, dependencies, associative semantics, and context information may be stored as well [Xu et. al 2003]. This supports a multi-dimensional and structured search of the registry; hence, discovery of the schemas is more efficient.

Placing a schema into one or more classifications can be a tedious and error-prone task. Placing a schema in a wrong node in a classification not only makes the schema less accessible but also has a risk of misinterpretation by other users. In addition, placing a schema in too generic a node makes the *Model Discovery (A1)* activity less efficient by inundating the user with too many schemas. To execute the placement correctly, a user must understand the semantics of the classification schemes as well as all of the schemas.

An envisioned tool to support the *Model Registration* activity is the classification assistant. This tool would use a semantic similarity measure to suggest classification nodes to the user. Measures that assume that terms and data structures with identical names have the same meaning regardless of context are identified in [Alspaugh et. al 1999, Do et. al 2002]. Measures based on properties and attributes are defined in [Peng et. al 2002], [Ryu and Eick 1998], [Schallehn and Sattler 2003], and [Dong et. al 2004]. Measures based on weighted property values and structures are defined in [Bhavsar et. al 2003].

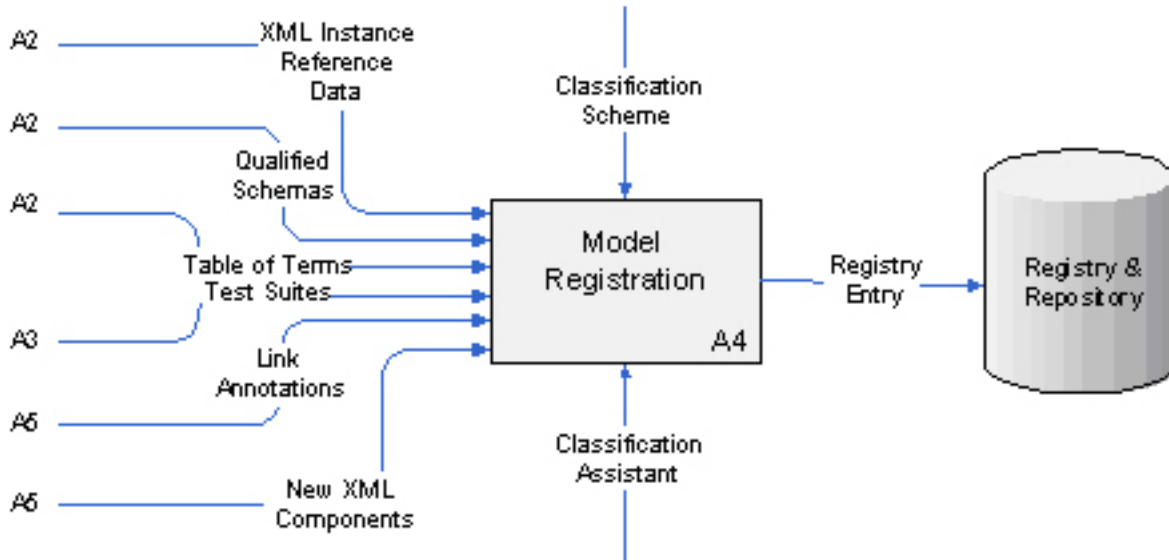


Figure 7. Activity A4 - Model Registration

3.5. Model Integration

The *Model Integration* activity ensures that new schemas and extensions are semantically coherent and, where possible, eliminates duplicate and overlapping schemas (see Figure 8). The first subactivity identifies terms and data structures that are semantic duplicates and/or overlaps (more on this below). Where possible, duplicates are eliminated by requesting changes to the original schemas as shown in A5.2. When elimination is not possible, such as when one or more of the schemas are already in use or it is a standard controlled by an outside party, cross-link annotations are created.

Similarly, in activity A5.3, the preferred approach to resolving overlaps would be to restructure and establish relationships within the schemas. When that is not possible, cross-links between the overlaps should be annotated to ensure that the relationships can be identified and managed. Annotation tools based on XML Linking Language (XLink) [XLink] and Resource Description Framework (RDF) [RDF] may be used.

Model Integration can be complex particularly when there is semantic ambiguity in the model or when part of the model needs to be restructured to accommodate a new relationship in the overlapping semantics. The tools we have conceptualized for the *Model Integration* activity include a *Semantic Similarity Measure* and a *Semantic Alignment Algorithm*. The semantic similarity measure provides assistance in the activity A5.1 described above and the *Model Registration (A4)*, while the semantic alignment algorithm supports activity A5.3. The semantic similarity measure assists in identifying the semantic duplication and overlaps by providing quantitative guidelines for assessing the semantic proximity of terms. The semantic alignment algorithm would (1) discover the relationships between the new terms or structures and the existing ones and (2) suggest changes to accommodate the new relationships. Ongoing research works such as [Stuckenschmidt and Visser 2000], [Ambite and Knoblock 1995], and others mentioned in A4 provide a basis for these two tools.

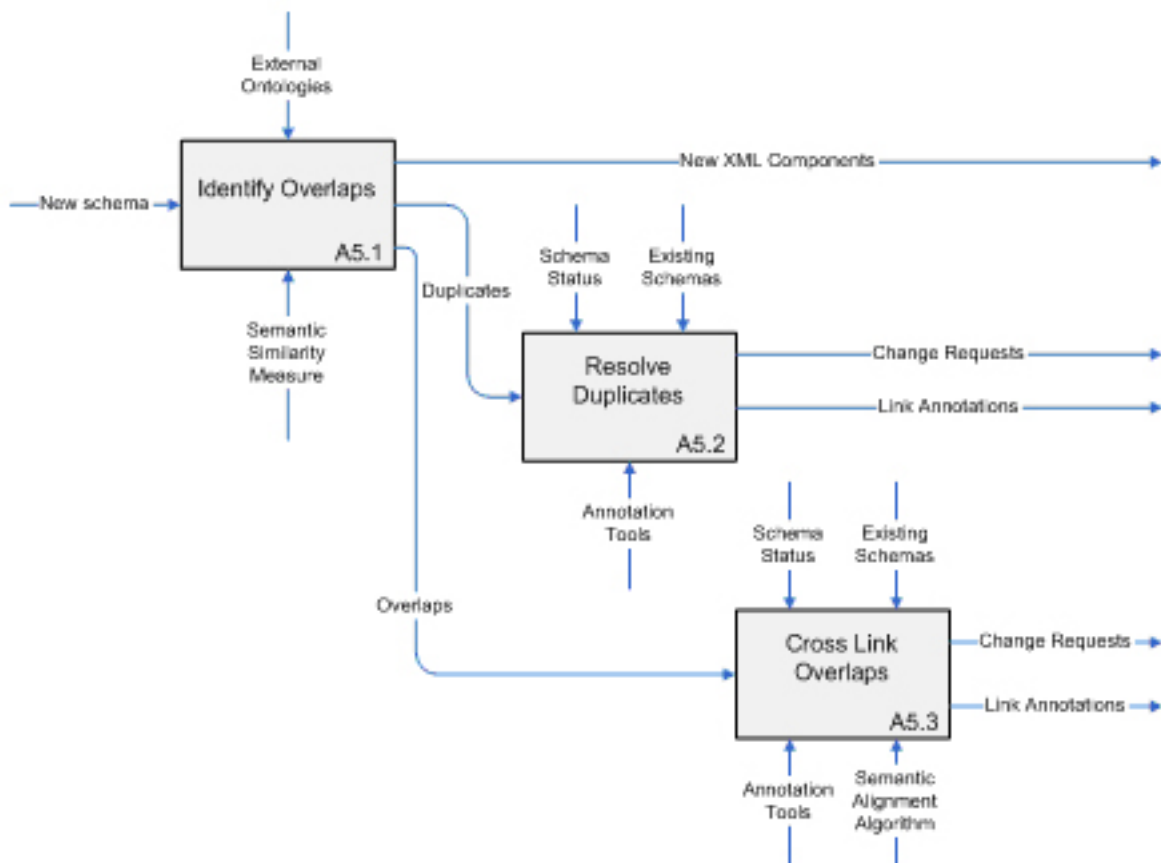


Figure 8. Activity A5 - Model Integration

4. Conclusion and Future Work

We have identified five high-level activities involved in creating and maintaining XML schemas used for systems integration. We also described a number of existing tools developed at NIST and elsewhere to support the execution of these activities. Finally, we outlined several semantic-based tools that have been conceptualized but not yet implemented.

We believe that these activities and tools are key to achieving coherence in the life cycle of a schema. Without these tools, the registry is nothing but a simple file store. Our future work will mainly focus on making the semantic tools a reality while continuing to generalize and improve the existing tools.

5. Product Disclaimer

Certain commercial software products are identified in this paper. This use does not imply approval or endorsement by NIST, nor does it imply that these products are necessarily the best available for the purpose.

6. Best Practices References

ASC X12C Communications and Controls Subcommittee (October 2002). ASC X12 Reference Model for XML Design. ASC X12C/2002-61

ebXML Technical Architecture Specification v1.0.4, 16 February 2001

KIEC (Korean Institute of Electronic Commerce) XML Guideline

Lockheed Martin Federal Systems (October 2002). Global Combat And Support System – Air Force BOD Developer's Guide Draft Version 1.1. Department of Air the Force Headquarters Materiel Systems Group (MSG). OASIS UBL Naming and Design Rules Subcommittee (November 2003). Universal Business Language (UBL) Naming and Design Rules.

Roger Costello XML Schemas: Best Practices <http://www.xfront.com/BestPracticesHomepage.html>

Roger Costello XML Schema Versioning. <http://www.xfront.com/Versioning.pdf>

Rowell, M., Feblowitz, M. (2002). OAGIS 8 Design Document (Draft 0.93)

US Federal CIO Council Architecture and Infrastructure Committee, XML Working Group (April 2002). Draft Federal XML Developer's Guide.

Bibliography

[AEX] <http://www.fiatech.org/projects/idim/aex.htm>

[Alspaugh et al 1999] Alspaugh, T.A., Anton, A.I., Barnes, T., and Mott, B.W. An Integrated Scenario Management Strategy, In Proceedings of the 4th IEEE Symposium on Requirements Engineering (RE99), Limerick, Ireland, pp. 142-149, 1999

[Ambite and Knoblock 1995] Ambite, J.L. and Knoblock, C.A., Reconciling distributed information sources. In Working Notes of the AAAI Spring Symposium on Information Gathering in Distributed Heterogeneous Environments, Palo Alto, CA (1995).

[B2BTestbed] <http://www.mel.nist.gov/msid/b2btestbed/>

[Bhavsar et. al 2003] Bhavsar, V.C., Boley, H., and Yang, L. A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in e-Business Environments, In Proceedings of Business Agents and the Semantic Web (BASeWEB) Workshop, Halifax, Nova Scotia, Canada, June 14 2003

[BSR] Basic Semantic Repository (Basic Semantic Registry Basic Reference Document, International Standard Organization). http://www.dcnicn.com/lamp/cals_97f/task11/html/Fbsr_undef/Repos-04.htm#P168_12909
[http://www.dcnicn.com/lamp/cals_97f/task11/html/Fbsr_undef/Repos-04.htm#P168_12909]

[CCT] http://www.mel.nist.gov/msid/b2btestbed/semantic_checking.html
[http://www.mel.nist.gov/msid/b2btestbed/semantic_checking.html]

[CSWizard] Schematron Editor Tool, <http://www.sf.net/projects/cs-wizard>

[Do et. al 2002] Do, H.H., Melnik, S., and Rahm, E., Comparison of Schema Matching Evaluations, In Proceedings of GI-Workshop Web and Databases, Erfurt, October, 2002.

[Dong et. al 2004] Dong, X., Halevy, A., Madhavan, J., Nemes, E., and Zhang, J. Similarity Search for Web Services, In Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.

[EbCC] DISA UN/CEFACT EbXML Core Component Specification version 1.9, . http://webster.disa.org/cefactgroups/tmg/downloads/CCWG/for_review/CCTS_V_1pt90.zip
[http://webster.disa.org/cefactgroups/tmg/downloads/CCWG/for_review/CCTS_V_1pt90.zip]

- [Schallehn and Sattler 2003] Federal Enterprise Architecture, <http://www.feapmo.gov/>
- [FEA] Schallehn, E., and Sattler, K.U., Using Similarity-based Operations for Resolving Data-Level Conflicts, In Proceedings of BNCOD'03, Coventry, pp. 172-189, 2003
- [Friedman-Hill 2002] Friedman-Hill, E., Jess the Rule Engine for the Java™ Platform Version 6.0a8. <http://herzberg.ca.sandia.gov/jess/>
- [Goyal 2004] Goyal, P. (2003), An XML Naming Assister for Elements and Types. National Institute of Standards and Technology Internal Publication, NISTIR 7143.
- [HyperModel] XMLmodeling.com, HyperModel, <http://www.xmlmodeling.com/>
- [IDEF0] Integration Definition for Function Modeling (IDEF0), Draft Federal Information Processing Standards Publication 183, National Institute of Standards and Technology, December 1993. <http://www.idef.com>
- [Ivezic et. al 2004] Ivezic, N., Kulvatunyou, B.S., Jones, A.T., Frechette, S., Cho, H., and Jeong, B. (October 2004), An Interoperability Testing Study: Automotive Inventory Visibility and Interoperability. Fourteenth E-Challenge Conference, Vienna, Austria.
- [ISO2003] ISO/IEC 19757-2:2003 Information technology -- Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG; RELAX NG; ISO - International Organization for Standardization. <http://relaxng.org>
- [ISO2004] Metadata Standards Organization Web Site, accessed May 2004. ISO/IEC 11179, Information Technology - Metadata Registries (MDR). <http://metadata-stds.org/> .
- [Jelliffe 2003] Jelliffe, R., The Schematron Assertion Language 1.5, Academia Sinica Computing Center. <http://www.ascc.net/xml/resource/schematron/Schematron2.000.html>
[<http://www.ascc.net/xml/resource/schematron/Schematron2000.html>]
- [Kulvatunyou 2004] Kulvatunyou, B.S., Ivezic, N., Jeong, B. (July 2004), Testing Requirements to Manage Data Exchange Specifications in Enterprise Integration - A Schema Design Quality Focus. The 8th World Multi-conference on Systemics, Cybernetics, and Informatics, Orlando, FL, USA.
- [OAG] Open Application Groups Web Site, accessed August 2004. Open Application Group home page. <http://www.openapplications.org>
- [OAGIS] Open Application Groups Web Site, accessed March 2002. Open Application Group Integration Specification version 8.0. <http://www.openapplications.org/downloads>
- [OASIS] OASIS Conformance TC conformance resource, accessed August 2003, Glossary of Conformance Terminology, <http://www.oasis-open.org/committees/ioc/glossary.htm>
- [Peng et. al 2002] Peng, Y., Zou, Y., Luan, X., Ivezic, N., Gruninger, M., and Jones, A., Towards semantic-based integration for e-business, International Symposium on manufacturing and Applications, Orlando, FL (June 2002).
- [Ryu and Eick 1998] Ryu, T.W., and Eick, C.F., Similarity Measures for Multi-valued Attributes for Database Clustering, In Proceedings of Smart Engineering System Design Neural Network, Fuzzy Logic, Evolutionary Programming, Data Mining and Rough Sets (ANNIE'98), St. Louis, Missouri, USA, November 1-4, 1998
- [RDF] World Wide Web Consortium. RDF Specification Development page. Internet Web Site, <http://www.w3.org/RDF/#specs>

- [STAR2003] STAR (Standards for Technology in Automotive Retail). Making the Case for IT Standards in Retail Automotive. STAR publication, 2003.
- [Stuckenschmidt and Visser 2000] Stuckenschmidt, H. and Visser, U., Semantic translation based on approximate reclassification. Proceedings of the Workshop "Semantic Approximation," Granularity and Vagueness, KR'00 (2000).
- [TBG17] UN/CEFACT International Trade and Business Group Work Group 17 Web site, accessed August 2004. TBG 17 Harmonisation. http://www.disa.org/cefact-groups/tbg/wg/tbg17_main.cfm
- [Xlink] World Wide Web Consortium. XML Linking Language Recommendation 1.0 <http://www.w3.org/RF/2001/REC-xlink-20010626/>
- [Xu et. al 2003] Xu, Z., Karlsson, M., Tang, C., and Karamanolis, C. Towards a Semantic-Aware File Store, In Proceedings of HotOS IX: The 9th Workshop on Hot Topics in Operating Systems, Lihue, Hawaii, USA, May 18-21, 2003
- [XPath] World Wide Web Consortium. XML PATH Language Version 1.0 (November 1999). <http://www.w3.org/TR/xpath>
- [XSD] XML Schema, <http://www.w3.org/2001/XMLSchema.xsd>
- [XSDDOC] Bluetetra Software, XSDdoc 2.0 <http://www.bluetetra.com/xsddoc/>
- [XSLT] World Wide Web Consortium. XSL Transformations (XSLT) Version 1.0 (November 1999). <http://www.w3.org/TR/xslt>

Biography

Serm Kulvatunyou

[National Institute of Standards and Technology](http://www.nist.gov) [http://www.nist.gov]
Gaithersburg
Maryland
United States of America
serm@nist.gov

Serm is a principle architect of the Manufacturing B2B Testbed at the National Institute of Standards and Technology. He has helped the auto and the capital facility industry in XML standards development and conformance testing. He has also contributed to the EbXML BP and CC standards.

Katherine Morris

[National Institute of Standards and Technology](http://www.nist.gov) [http://www.nist.gov]
Gaithersburg
Maryland
United States of America
kcm@nist.gov

KC Morris is a computer scientist in the Manufacturing Systems Integration Division of the National Institute of Standards and Technology. Her work involves methods and mechanisms for supporting the specification and development of information interface standards and the testing of those standards. KC has been a technical contributor to ISO working groups and to several industrial consortiums, including FIATECH, PDES, Inc. and the National Industrial Information Infrastructure Protocols Program (NIIP). Currently, she is engaged in a project to develop and test XML Schema-based interfaces for the capital facilities industry. Earlier in her career she worked

with STEP, a.k.a. ISO 10303 The Standard for the Exchange of Product Model Data. She was a primary contributor to the development and validation of STEP's early implementation and testing mechanisms.

Buhwan Jeong

[National Institute of Standards and Technology](http://www.nist.gov) [<http://www.nist.gov>]

Gaithersburg

Maryland

United States of America

bjeong@nist.gov

Puja Goyal

[National Institute of Standards and Technology](http://www.nist.gov) [<http://www.nist.gov>]

Gaithersburg

Maryland

United States of America

pgoyal@nist.gov

Puja Goyal is a computer scientist in the Manufacturing Systems Integration Division of the National Institute of Standards and Technology. She has been involved with the AEX Testbed, developing tools to support XML Schema validation and naming conventions.