# A Validation Architecture for Advanced Interoperability Provisioning

Nenad IVEZIC[1], Ed BARKMEYER[1], Serm KULVATUNYOU[1], Albert JONES[1], Pat SNACK[2], Zoran MARJANOVIC[3], Hyunbo CHO[4]
[1]*NIST, 100 Bureau Drive, Gaithersburg, MD, 20899, USA*
*Tel: +001 301 975-3536, Fax: + 001 301 975 4482,*
*Email: {nivezic| edbark|serm|ajones@nist.gov}*
[2]*Automotive Industry Action Group, 26200 Lahser Rd., St. 200, Southfield, MI 48034, USA*
*Tel: +001 248 358-3570, Fax: +001 248 358-3570,*
*Email: {psnack@aiag.org}*
[3]*Faculty of Organizational Sciences, Jove Ilica 154, 11000 Belgrade, Serbia & Montenegro*
*Tel: +381 11 395-0800, Fax: +381 11 461-221,*
*Email: {marjanovic.zoran@fon.bg.ac.yu}*
[4]*Department of Industrial and Management Engineering, Pohang University of Science and Technology, San 31, Hyoja, Pohang 790-784, Korea*
*Tel: +82 54 279-2204, Fax: +82 54 279-2870,*
*Email: {hcho@postech.ac.kr}*

**Abstract:** We describe an architecture to validate new tools in support of interoperability provisioning processes that take place within industrial communities. We start from a best-practices standards development process where an industry community develops interoperability specifications that are used by vendors to make their software applications compliant to the community's requirements. We show how these standards-based interoperability provisioning processes may be enhanced using the new tools. We describe potential impact of these tools and discuss how that impact may be validated within the proposed architecture. We conclude by discussing the business value of the validation analyses discussed in this paper.

## 1. Introduction

As advanced interoperability technologies continue to be developed, the need to validate the resulting tools within industrial situations also continues to grow[1]. Among these situations, a particularly challenging class is the *community-driven enterprise application interoperability provisioning.* Within this class of *interoperability provisioning processes*, a global automotive, chemical, electronics, or another industry group (that comprise users, application vendors, and other stakeholders) engages its stakeholders to jointly develop and adopt interoperability artefacts that are implemented by the application vendors and, ultimately, tested for conformance and later deployment in the users' environments. Interoperability artefacts include business process models, data exchange models, and data exchange message schema specifications.

Presently, interoperability artefacts are represented for sharing at the implementation level as technical specifications. For example, an XML Schema-based business document specification devised by an industry community would be shared for implementation with the application vendors with presence in the industry. Other artefacts, such as business process models and data models that define the interoperability context, are exchanged as informative documents only. However, representing and sharing interoperability specifications at modeling levels may provide automation and cost-saving opportunities.

[1] This paper is based on the initial phase of the ATHENA Sub-Project B5.10: Piloting Including Technology Testing Coordination – Inventory Visibility and Interoperability (IV&I). The primary purpose of the sub-project is to perform validation of the ATHENA results (i.e., advanced tools and methodologies) on an industrial inventory visibility scenario.

## 2. Objectives

The objective of this paper is to describe a validation architecture that can perform assessments of novel tools in the context of community-based enterprise application interoperability provisioning. We focus on tools that create and use *model-based interoperability artefacts* such as enterprise modelling, ontology authoring, ontology-based annotation, and ontology-based reconciliation tools. Our validation architecture is designed to enhance the traditional standards development process by making use of the new tools' capabilities that support the activities within that process. We also describe the potential impact of those tools and provide ways of measuring that impact.

## 3. Methodology

We view the community-driven, enterprise-application interoperability provisioning to be an engineering process with a degree of automated support for some of the activities in that process. The goal of the automated support is to enable meaningful information exchange among a specific class of applications and within a particular application context. Typically, both are defined within the standards development process adopted by such a community. It is during such a process, that the initial business case and requirements are refined in the form of models and specifications that ultimately drive the implementation. These models and specifications, when correctly interpreted, provide descriptions of the target context and class of applications.

Additional criteria may be of interest when applying the automated tools. One criterion may be to minimize human intervention when a new application is substituted for an existing one. Another criterion may be to minimize the time required for adoption of an interface standard and its implementation in commercial applications. Other criteria include enhancement of the maintainability and quality of the standard [4, 5].

For our purpose, we assume the following key phases in the standards development processes: (1) modeling; (2) implementation; (3) testing; and (4) execution. These generic phases are the basis for our validation architecture and process. Using the Inventory Visibility and Interoperability (IV&I) community-driven interoperability provisioning approach as an example, Figure 1 shows key activities within the above phases [1].

During the Modeling Phase, the community initially takes part in the Business Process Modeling and Requirements Gathering activity where it adopts a modeling approach and, based on its collective domain knowledge and legacy models, produces Business Process Models and Data Exchange Requirements. Next, the community engages in the Data Exchange Modeling activity where it takes the results of the previous activity, assumes a standard industry terminology (e.g., OAGIS [2]), and a data modeling approach to detail the data exchange models for the business process. While a data modeling tool may be adopted, rigour is not usually a strong objective at this time.

Next, the community enters the Business Object Document (BOD) Schema and Guidelines Definition activity where it refines the data exchange model. This model, which uses XML Schema specification, defines the actual messages to be exchanged among the trading partners [2]. At this time, the corresponding message guideline documents are created detailing the structure of the schema specification. That structure includes element inclusions and exclusions, optional and mandatory element specifications, conditional relationships among element definitions and/or element values, and, the use of standard codes.

Finally, the Messaging Infrastructure Architecting activity specifies the infrastructure on which the actual messages will be executed and exchanged. Presently, this activity is based on either ebXML [6] or Web Services [7].

At the Implementation Phase, the model specifications are turned over to the application vendors. During the Application Vendor's Specifications Check, the vendors determine whether the specifications from the previous phase are adequate to complete the implementation phase. Next, during the Application BOD Interface Enablement activity, each vendor implements the BOD specifications and makes the application BOD-enabled. During Messaging Adoption activity, the vendor adopts a messaging execution solution based on ebXML, Web Services, or both.

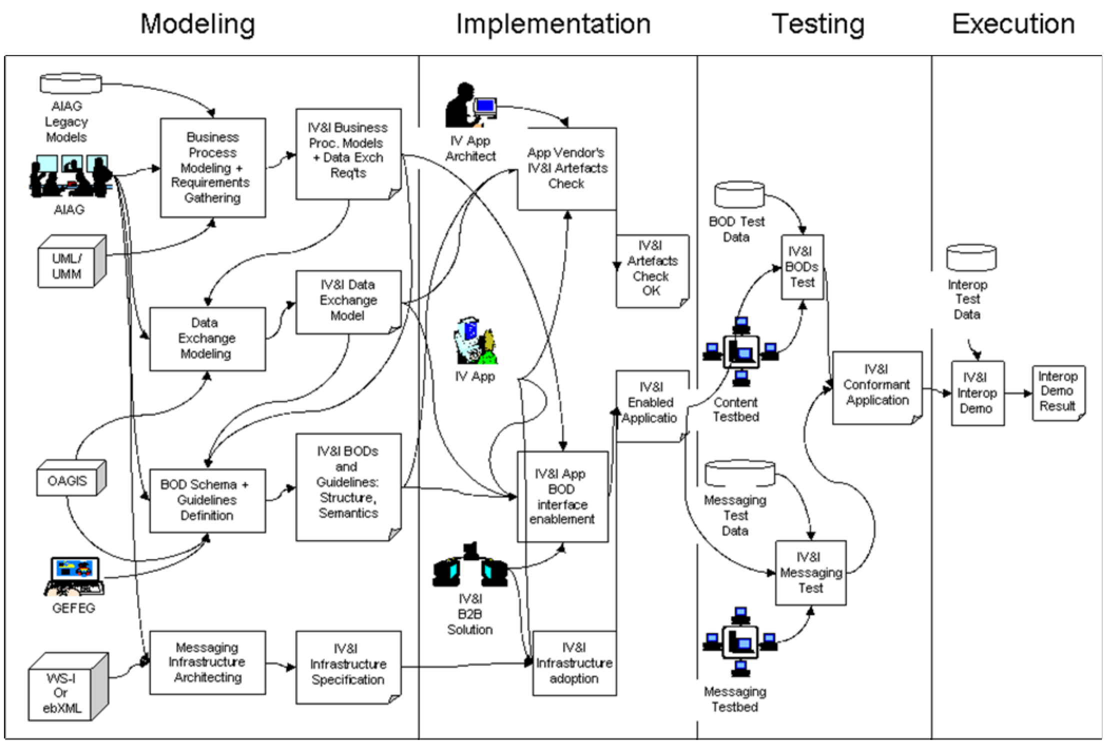


*Figure 1: A Best-Practices Standards Development Process*

At the Testing Phase, two, essential, conformance tests must be performed. First, the BOD Conformance Testing activity engages each application vendor to demonstrate conformance of its BOD implementation to the specifications. Next, the Messaging Testing activity engages the vendors to demonstrate conformance of their messaging execution solutions to the messaging specifications. A neutral testing facility is often used to execute these tests in collaboration with the vendors. At the end of the phase, a vendor can claim an IV&I conformant application.

Finally, at the Execution Phase, interoperability claims are tested in proof-of-concept demonstrations and industry pilots. Both are based on testing scenarios and test data that mirror real industrial concerns and requirements. During actual deployment efforts, additional interoperability tests may be needed to mirror exact requirements from specific customers.

Collectively, these activities, actors, and outcomes represent the basis for our adopted validation architecture, which we discuss in the next section.

## 4. Architecture Details

In this paper, we discuss two perspectives of the proposed validation architecture: an integral one and a semantic-interoperability one. The integral perspective describes the end-to-end data exchange scenario that produces and then utilizes model-based interoperability artefacts and the novel tools used in that process. The semantic-interoperability perspective is focused on implementing BODs within the applications in the Implementation Phase.

### *4.1 Validation Architecture: An Integral Perspective*

The integral perspective is illustrated in Figure 2 for the case of the IV&I project. Within this perspective, the tools that create and use *model-based interoperability artefacts* affect the IV&I process in the modeling, implementation, and execution phases.

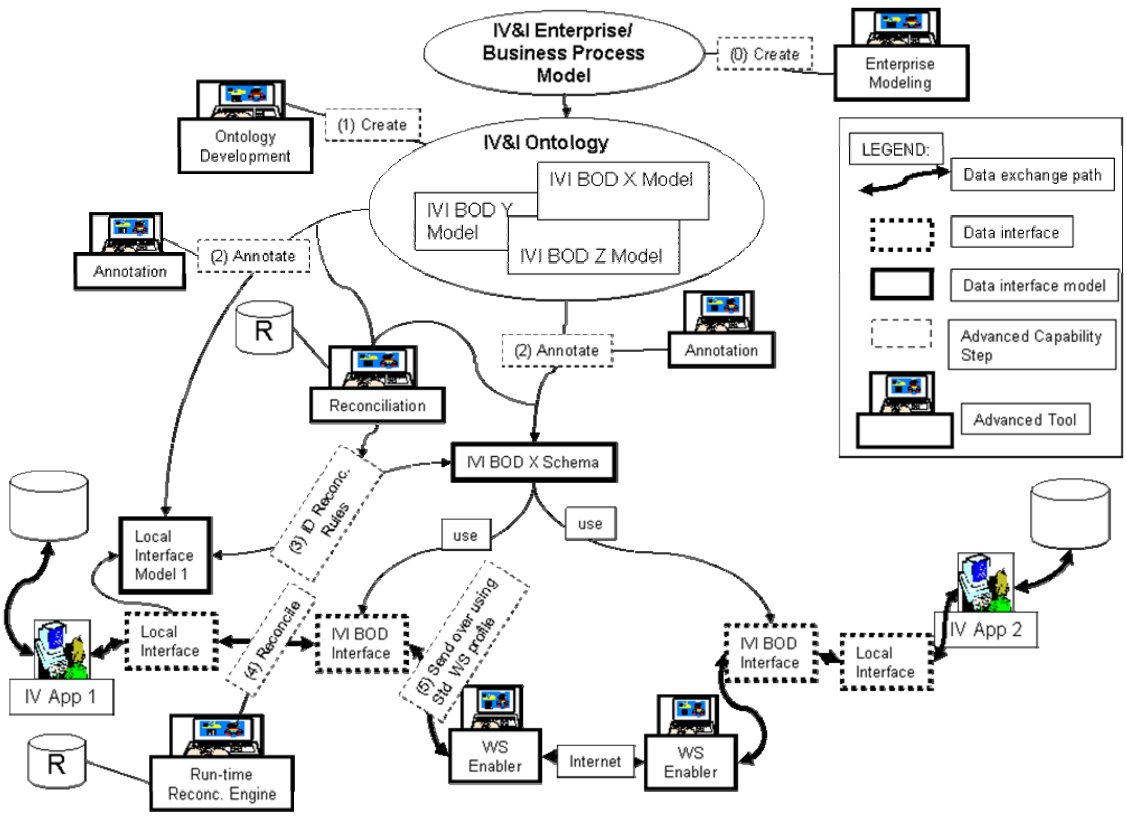


*Figure 2: An Integral Perspective on the Validation Architecture*

The first two steps make up the modeling phase of the process. In Step 0, an Enterprise Modeling tool is used to create the IV&I Enterprise/Business Process Model. In Step 1, an ontology modeling tool is used to refine the concepts in enterprise and business process model to the level needed to define the intended data exchange messages. The result is the IV&I Reference Ontology (RO).

Steps 2 and 3 make up the implementation phase. In Step 2, an Annotation Tool is used with the RO by application vendors to annotate their local interface models and by the IV&I developers to annotate the IV&I BOD Schemas. In Step 3, these annotations are used by a Reconciliation Tool to resolve differences between the local application modes and IV&I models. This tool generates a rule base (R) that translates data instances from a source-application data format to the intermediate IV&I BOD data format and, then, to the target-application data format. The actual reconciliations make use of the RO.

Steps 4 and 5 make up the Execution Phase. In step 4, the Run-time Reconciliation Engine uses the rule base (R) to translate the source application data format to the intermediate IV&I data format. This transformation may be done locally or using a globally accessible reconciliation service. In a business-to-business environment, the transformation is likely done locally in order the keep proprietary information private. In such case, data are exchanged using a BOD data format. In Step 5, the IVI BOD instance data is transferred using a standard service profile by the Web Service Enabler Tool.

The integral perspective enables us to conduct experiments and make measurements that address the ultimate objective: an interoperable end-to-end data exchange. Two proposed measures are related to quality of, and efficiency to, produce interface implementations. The quality of the implementation is measured through BOD implementation testing methods such as those described in [3]. Such methods are used to determine whether the data exchange specification has been interpreted and mapped onto the local data elements correctly. The efficiency to produce interface implementation is measured by comparing the time required for a vendor to implement a BOD (over a number of BOD specifications) using the advanced and the traditional approach.

The next section gives a detailed perspective on the use of semantic-based technologies during implementation phase where we also refine the efficiency measurement concept.

*4.2 Validation Architecture: A Semantic-based Interoperability Perspective*

A semantic-based interoperability perspective is illustrated in Figure 3 for the BOD implementation steps of the Implementation Phase (described in Section 4.1). We assume the IV&I Reference Ontology exists and that it correctly represents the concepts necessary to describe the intended data exchange.

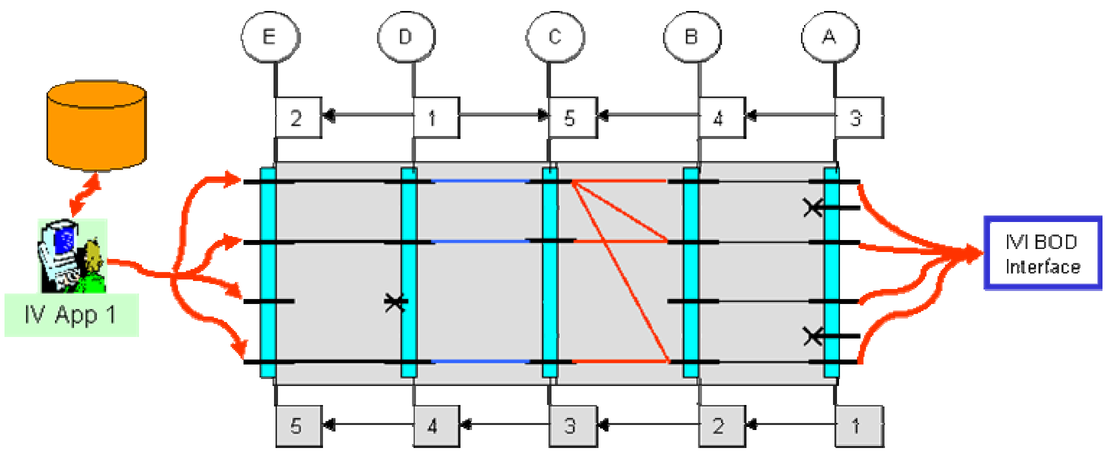


*Figure 3: The 'Virtual Data Transformation Pipe'*

The figure illustrates steps (A-E) encountered during BOD enablement for the As-Is and the To-Be scenarios. The As-Is scenario is the conventional BOD implementation method and is based on the *traditional approaches* while the To-Be scenario is based on the *advanced approaches* enabled by the new tools that enhance BOD implementation shown in Figure 2. Each step is associated with a vendor's action to make the application support the data exchange specification. Collectively, these steps form a 'virtual data transformation pipe' (implicit in a BOD interface implementation) that the application data goes through en route to the receiving application.

In the As-Is scenario, the steps (marked with numerals at the bottom of the figure) are performed by the application vendor to implement an IV&I conformant interface (based on the IV&I BOD XML Schema-based specification). These steps are defined below.

1. Adopt the <u>IV&I BOD XML Schema-based Specification</u> as the data exchange specification (as designed and provided by the IV&I community). Each data element (represented by the 'horizontal bar' symbol) of the intended exchange is defined by an associated data type.

2. Create the IV&I BOD Interpretation from the IV&I BOD usage guidelines document that defines the meaning of the BOD data exchange specification (e.g., by including and excluding data elements, specifying mandatory and optional data elements, and defining the semantics of these elements) using informal, human readable statements. (Note that some elements of the BOD Specification may not be used, as defined by guidelines and indicated by the crossed-out element symbols in Figure 3.)
3. Create a Local Data Repository Data Type by identifying the data elements from the local application data repository matching the IV&I BOD Interpretation elements. (In this step, the vendor follows the BOD Interpretation to identify the local data elements of matching types and meanings. This step may be supported by a local data dictionary or a conceptual data model guiding the implementer in selecting and validating the data types.) At this point, a manual semantic mapping is performed for each element of the Local Data Repository Data Type that is mapped onto IV&I BOD Interpretation: these local data elements are aggregated, transformed, and/or extracted from, in order to map onto the selected IV&I BOD Interpretation elements.
4. Formulate a Local Data Access Query based on the IV&I BOD Interpretation and the constructed Local Data Repository Data Type. Obtain, as a by-product, 1-to-1 mapping between the Local Data Access Query and the Local Data Repository Data Type – allowing direct access/store of the query result in the constructed local data type.
5. Use a Local Data Access Specification (e.g., the local data repository access application programming interface) to formulate the Local Data Access Query.

In the To-Be scenario, the steps (marked with numerals at the top of the figure) are taken by the application vendor based on the IV&I BOD XML Schema-based specification annotated using the IV&I Reference Ontology and enabled by the advanced tools that are subject of our validation. These steps are defined below.

1. Create an Annotated Local Data Repository Model based on the IV&I Reference Ontology (IV&I RO) published by the IV&I community. The vendor uses the local data dictionary and local repository conceptual model to identify the local data elements that correspond to the IV&I RO elements and to annotate the data elements with these IV&I RO elements. Note that this step is driven by the IV&I Reference Ontology, not by a specific IV&I BOD interface specification. One may think of this step occurring during the IV&I modeling time when the vendor works with the IV&I community to align its application with the intended IV&I semantics. (Hence, this step is a part of the Modeling Phase illustrating the subtle point about a potential impact of ontology-based approach where a traditionally implementation step moves to the modeling phase.)
2. Define a mapping from the Annotated Local Data Repository Model to the Local Data Access Specification. In this step, the vendor defines the manner in which the local data repository is accessed for each relevant element of the annotated repository model (e.g., using a local repository application programmer interface). (This step, as the previous, can be completed during Modeling Phase.)
3. Adopt the IV&I BOD XML Schema Specification as the data exchange specification (as in the As-Is scenario).
4. Adopt the IV&I BOD Annotation (as provided by the IV&I community). The annotated BOD contains references to the IV&I Reference Ontology and provides a common, formal interpretation basis of the data exchange specification.
5. Create a Local Interface Model by selecting relevant elements from the Annotated Local Data Repository Model to match the IV&I BOD Annotation. Note that the application vendor has at disposal both the IV&I BOD Annotation and the Annotated Local Data Repository Model, both annotated using the IV&I RO. When a specific IV&I BOD Annotation is presented, it informs the selection of relevant Annotated Local Data Repository Model elements. In this manner, we obtain a mapping from the

Annotated Local Data Repository Model to the Local Interface Model. Finally, the vendor creates Reconciliation Rules (using Tool-supported Semantic Mapping) between the Local Interface Model and the IV&I BOD Annotation. The reconciliation rules are created using the tool described in Section 4.1).

These descriptions of the As-Is and To-Be steps allow reasoning about the validation issues and metrics – the next section will give an example of such validation metrics.

## 5. Results

The discussion in the last section allows qualitative reasoning about the advanced tool validation metrics. For example, descriptions of the As-Is and To-Be steps, give an "improvement in efficiency of implementation" measure ($M^{efficiency}$) as the difference in times for a vendor to implement a BOD using the *advanced* and *traditional approaches*:

$$M^{efficiency} = T^{Advanced} - T^{Traditional}$$
$$T^{Traditional} = T^{T}_{B\text{-}A} + T^{T}_{C\text{-}B} + T^{T}_{D\text{-}C} + T^{T}_{E\text{-}D}$$
$$T^{Advanced} = T^{A}_{E\text{-}D} + T^{A}_{B\text{-}A} + T^{A}_{D\text{-}B}$$

where

- $T^{T}_{B\text{-}A}$ – As-Is time to interpret the BOD specification based on the guidelines.
- $T^{T}_{C\text{-}B}$ – As-Is time (a) to identify the data elements from the application's local data repository matching the IV&I BOD Interpretation; and (b) to aggregate, transform, and/or extract data to map the local data onto IV&I BOD Interpretation elements.
- $T^{T}_{D\text{-}C}$ – As-Is time to define local data access query based on the local data elements identified in previous step.
- $T^{T}_{E\text{-}D}$ – As-Is time to formulate the Local Data Access Query using the Local Data Access Specification.
- $T^{A}_{E\text{-}D}$ – To-Be time (a) to create an Annotated Local Data Repository Model and (b) to map from the Annotated Local Data Repository to the Local Data Access Specification.
- $T^{A}_{B\text{-}A}$ – To-Be time to interpret the BOD specification based on the RO-based annotations.
- $T^{A}_{D\text{-}B}$ – To-Be time (a) to select relevant elements from the Annotated Local Data Repository Model, as described in the IV&I BOD Annotation, and create the Local Interface Model; and (b) to create reconciliation rules between the Local Interface Model and the IV&I BOD Annotation.

Assuming (in the To-Be scenario) creation of an annotated local repository model during Modeling Phase, the first time factor $T^{A}_{E\text{-}D}$ effectively goes to 0 at Implementation Phase. Then, the following measurements of interest may be performed:

(1) Compare the time $T^{T}_{B\text{-}A}$ using the traditional approach required to interpret the BOD specification based on the guidelines to the time $T^{A}_{B\text{-}A}$ required to interpret the specification based on the RO-based annotations.

(2) Compare the time $T^{T}_{C\text{-}B} + T^{T}_{D\text{-}C} + T^{T}_{E\text{-}D}$ required using the traditional approach to identify, assemble and/or extract, map onto the BOD interpretation, and express in terms of the local data repository access specification, the local data elements – to the time $T^{A}_{D\text{-}B}$ required using the advanced approach to select relevant elements from the annotated local data repository model and to reconcile rules between the local model and the IV&I BOD annotation.

We conclude from these analyses, that the advanced semantic-based interoperability approaches is likely to be beneficial if (1) the BOD specification is complex, (2) the BOD specification is changing quickly and requires maintenance, or (3) there are many BODs to be managed that reuse many specification components.

## 6. Business Case

The following are two examples of business value of the validation architecture and its use:

- The validation architecture, because it uses the standards development processes as its basis, provides a roadmap for the 'next generation' interoperability standards development processes. Supported by eventual validation experiments (to follow from the architecture), one may design and implement a new standards development process with greater certainty in the efficiency and quality of results than otherwise possible.
- An analysis result that follows easily from the proposed validation architecture is that there is a potential to reduce dramatically the time-to-market for the vendor companies that are successful in using advanced technologies to move several implementation tasks to the modeling phase.

## 7. Conclusions and Future Work

In this paper, we described a validation architecture in which emerging tools for *community-driven enterprise application interoperability provisioning* may be applied and where the potential effect of these tools may be demonstrated and measured.

Our architecture is based on the traditional standards development processes. It makes use of emerging tools to enhance end-to-end data exchange. We described two perspectives on that architecture: an integrated perspective and a semantic-based interoperability perspective. We illustrated a qualitative validation measure that builds on the detailed semantic-based interoperability perspective for BOD enablement. Our future work will validate a number of novel tools using the proposed architecture and the IV&I scenario.

## 8. Disclaimer

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

## References


[1] Automotive Industry Action Group (AIAG) Web Site, accessed April 2005. Available at http://www.aiag.org/

[2] Open Application Groups Web Site, accessed April 2005. Available at http://www.openapplications.org.

[3] Kulvatunyou, B.S., Ivezic, N., Jones, A.T.: Content-Level Conformance Testing: An Information Mapping Case Study. In Khendek F., Dssouli, R. (eds.): Testing of Communication Systems. Lect. Notes in Comp. Science, Vol. 3502. Springer-Verlag (2005).

[4] Kulvatunyou, B.S., Morris, K.C., Buhwan, J., Goyal, P.: Development Life Cycle and Tools for XML Content Models. XML Conference, November 2004, Washington, DC.

[5] Kulvatunyou, B., Ivezic, N., Buhwan J.: Testing Requirements to Manage Data Exchange Specifications in Enterprise Integration – A Schema Design Quality Focus. 8th World Multiconference on Systemics, Cybernetics and Informatics (SCI), July 2004, Orlando, Florida.

[6] OASIS ebXML Messaging Services Specification Technical Committee Web Site. EbXML messaging service specification version 2.0. http://www.oasis-open.org/committees/documents.php?wg_abbrev=ebxml-msg.

[7] WS-I Organization Website (Accessed December 2005), Online available via http://www.ws-i.org/.