Standards-Based Semantic Middleware

Nenad IVEZIC[1], Serm KULVATUNYOU[1], Marko VUJASINOVIC[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|serm|markov@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*
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 a novel semantics-based architecture for standards-based approaches to interoperable applications. The paper includes our experiences in prototyping the proposed architecture and demonstrating the resulting semantic middleware capability. This capability enabled legacy applications with heterogeneous interfaces to communicate in an interoperable fashion. An industrial scenario was used to showcase the new features of the architecture.

1. Introduction

Today, standards-based approaches are the most commonly used means to achieve large-scale, interoperable, business-to-business (B2B) applications both within and across industry sectors. Accordingly, an ability to affect significantly the efficiency and quality of *standards-based approaches to interoperable applications* will have profound consequences for industry verticals, cross-industry value chains, and, the bottom line of the enterprises.

In this paper, we describe an innovative, semantics-based architecture for standards-based approaches to interoperable applications within the B2B context. This novel architecture was developed within the recently completed EU-funded ATHENA project [1]. In a previous paper, we outlined the envisioned architecture that accommodates industry requirements for B2B standards while utilizing the potential of a new semantics-based approach [2]. Presently, we summarize the results of an initial implementation of this semantics-based architecture that gave rise both to a novel standards development process and a semantic middleware capability.

2. Objectives

The objective of this paper is to present our experience in prototyping a novel, semantics-based architecture for standards-based approaches to interoperable applications. We discuss an assessment of this architecture where we utilize its semantic middleware capability to enable interoperable applications within a realistic industrial scenario.

3. Methodology

Presently, industry consortia use mixtures of semi-formal and informal methods when utilizing *standards-based approaches to interoperable applications* in the B2B context; however, these approaches are error-prone, labor-intensive, and lead to ambiguous specifications when capturing the intended meaning of the targeted data exchange. Such approaches to interoperability may be summarized at the following three levels

- At the *B2B enterprise modeling level*, an industry consortium works to define the target Business Process Model (BPM). This model, however, is captured using informal, non-computational methods (i.e., free-form text and diagrams).
- At the *B2B message exchange design level*, an industry consortium adopts and extends the best-practice message schemas (e.g., based on XML Schema [3]) to come up with standard, data-exchange schemas. Also, at this level, the application vendors specify mappings between their applications' interface schemas and the standard data exchange schemas. The meaning of the schema elements and mappings between the elements of the applications' interface schemas and the elements of the standard data-exchange schemas, however, are captured using syntactic (i.e., data format definition) approaches and free-form text.
- At the *B2B message exchange implementation level*, the application interface-to-data-exchange schema mapping specifications are used as the specification to implement application standard-conformant data interfaces. These interface implementations are, however, manually intensive processes and vulnerable to errors as their meanings are inferred on the basis of data-format definitions and free-form text (as specified previously at the B2B message design level).

In the future, however, we would like to use novel semantics-based methods and enhance the *standards-based approaches to interoperable applications* in the B2B context. A new architecture is envisioned to enable the following advances at each of the three levels identified above [2]

- At the *B2B enterprise modeling level*, the fundamental concepts and relations may be captured using computer-processable Enterprise Models that comprise elements of the target BPM.
- At the *B2B message exchange design level*, the advanced approaches (1) may expose the data-exchange logic behind the application interface schemas and the standard data exchange schemas and (2) may define precise semantics of the mappings between the elements of the applications' interface schemas and the elements of the standard data-exchange schemas.
- At the *B2B message exchange implementation level*, the application standard-conformant data interfaces may be implemented automatically: the mappings defined at the *B2B message exchange design* level may be executed directly using semantic reconciliation engines. In this way, consistent, conformant, and automated implementations of the standard interfaces may be possible through logical schema reconciliation.

To validate and assess the proposed semantics-based architecture, we used an existing *standards-based approach to interoperable inventory visibility (IV) applications.* This existing standards-based effort, named Inventory Visibility and Interoperability (IV&I) at the Automotive Industry Action Group (AIAG), focused on the electronic Kanban (eKanban) business process that the inventory visibility applications were required to support [4]. Our semantics-based architecture anticipated the following three phases to enhance this existing standards-based approach and enable interoperable IV applications

- In the **First Phase** (P1), an Enterprise Modeling tool is brought to bear on the eKanban Business Process Model capture. A goal for the enterprise modeling process is to capture the eKaban process data exchange requirements for the communicating partners (i.e., supplier and customer). In the course of validating the architecture, we also wanted to show that multiple tools may exchange the enterprise modeling results. Next, the following modeling activity refines the captured business process model to the level of precise atomic and composite application domain concepts and relationships to define the intended meaning of the information exchange. As a result, an eKanban process Reference Ontology (RO) is developed using an ontology authoring tool.

- Then, in the **Second Phase (P2)**, two independent types of activities take place: (1) the application vendors annotate their respective local application interface schemas and (2) the eKanban messages specification team annotates the standard message (i.e., Business Object Document (BOD)) schemas. The annotations are done with respect to the RO developed in the First Phase. To accomplish the annotation process, a semantic annotation tool is used. Next, the annotations of the local and standard BOD schemas are used to specify reconciliation rules for the differences between (1) each of the local application interface schemas or (2) the BOD schemas, on one side, and the RO, on the other. A reconciliation tool is used to generate a rule base for each of the reconciliation situations: (1) data instance translation from a source application data format and terminology to the RO format and terminology (and back); and (2) data instance translation from the RO format and terminology to the standard eKanban BOD format and terminology (and back).
- Next, during the **Third Phase (P3)**, the run-time reconciliation engine is employed to use the reconciliation rule bases and to translate the eKanban message instances from the source application data format and terminology to the standard eKanban BOD data format and terminology. This transformation may be done locally or using a globally accessible reconciliation service. Finally, the transformed eKanban BOD instance data are sent 'over the wire' using a standard Web Service profile configured and executed by a Web Service execution tool.

Incidentally, this novel semantics-based architecture for standards-based approaches may also be seen to comprise two distinct activities (1) *the public interface model definition* where the industry consortium specifies the standard data exchange logic, terminology, and data format for the target business process; and (2) *the private-public interface models reconciliation* where the application vendors align their proprietary private interface models with the public models.

4. Technical Details

The ATHENA-enabled semantics-based architecture for standards-based approaches to interoperable applications was implemented as shown in Fig.1. The following ATHENA results and tools were used in support of the architecture [1]:

- MOOGO and ARIS Enterprise Modeling Tools to capture eKanban business process in a computable form.
- POP* Enterprise Model Exchange Specification and MPCE Enterprise Model Repository (not shown) to provide model management and exchange services;
- ATHOS Ontology Management and Authoring Tool to capture the IV&I (eKanban) Reference Ontology (RO).
- ASTAR Model Annotation Tool to annotate the application interface schemas and message (i.e., BOD) schemas.
- ARGOS Reconciliation Rules Authoring Tool to create reconciliation rule bases translating data instances from either (1) an application data format and terminology to the RO format and terminology and vice versa or (2) the IV&I data format and terminology to the RO format and terminology and vice versa.
- ARES Run-Time Reconciliation Engine to execute reconciliation rules as a run time service.
- THEMIS Model Management Repository (not shown) to manage the application interface and message schema models.

- Johnson Web Services Execution Tool to send standard messages 'over the wire'.

In addition, a number of tools were built either to implement the approach following the specific requirements from the automotive industry B2B context or to integrate the ATHENA tools

- Transformer tools include (1) XSD2RDFS transformer tool to enable the business applications that currently use XML Schema-based interfaces to use the ATHENA semantic mediation tools that expect RDF Schema-based representation [5]; and (2) at run time, XML2RDF (i.e., AXTOR) and RDF2XML (i.e., ARTOX) tools to transform XML message instances from and to a compatible RDF representation, in conformance to the XSD2RDFS transformation [6].
- Gateway to orchestrate, at run-time, uses of the Transformer tools, the ATHENA semantic reconciliation engine (ARES), and the Johnson WS execution engine so as to transform a proprietary message into an eKanban conformant message and to send it "over the wire."
- Web Services Security (WSS) plug-in to enable the Johnson WS execution engine to meet the WS-I security specifications as required by the automotive industry [7].

Also, we made use of two IV applications and an IV&I conformant application supporting the eKanban process

- The Apolon open source IV application with a proprietary eKanban messaging interface based on RDF Schema [8].
- GM experimental enterprise application with a proprietary eKanban messaging interface based on XML Schema.
- Ford Test Harness (FTH) application that implements XML Schema-based IV&I-conformant eKanban interface.

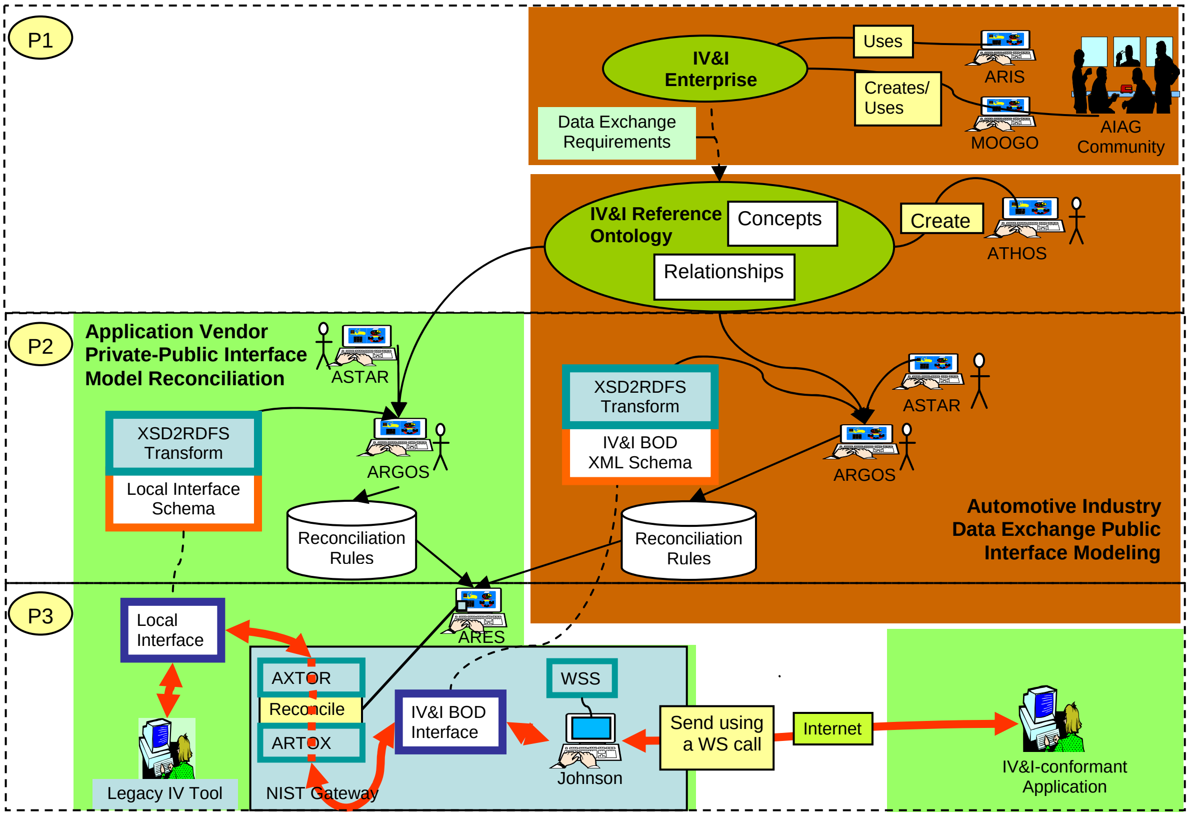

Figure 1: ATHENA-enabled Semantics-based Architecture for Standards-based Approaches to Interoperable Applications

The implemented semantics-based architecture was validated and was able to support interoperable applications by following these steps

- Initially, the enterprise modeler worked with the existing IV&I eKanban specifications to extract and define an IV&I eKanban Enterprise Model. The enterprise modeler used an EM tool (i.e., MOOGO) and an EM repository to author the IV&I eKanban Enterprise Model and capture the requirements for data exchange messages between the IV&I-conformant applications in a computable form. A part of the model was exchanged between multiple EM tools using the POP* and MPCE.
- The eKanban RO was created using the ATHOS ontology authoring tool by an ontology designer. The RO contains concept and relationship definitions necessary to create models of IV&I eKanban documents. These models of the documents were based on the automotive-industry-adopted data-exchange-schema standard and interviews done with the business process analysts from the industry.
- Next, based on the eKanban data exchange specification, the public interface model was defined; however, the data exchange schema specification was required to be represented using the Open Applications Group Integration Standard (OAGIS) XML Schema-based BODs [9]. These OAGIS XML Schemas were to be annotated and reconciled with the RO using ASTAR and ARGOS tools. To accomplish this, however, the XML Schemas had to be transformed into the required RDFS format using the XSD2RDFS tool. The outcome of these activities is a set of Reconciliation Rules that effectively maps BOD message instances, transformed into the required RDF format, into and out of the RO representation. In this way, the Public Interface is modelled. Significantly, the vendors do not need to deal with the Public Interface data-exchange schemas (i.e., the eKanban BOD schemas) – the vendors need not do anything specific about the particular public interface schemas.
- Then, the private-public interface model reconciliation was completed; the ASTAR and ARGOS tools are used in a similar manner to reconcile the public BOD and private interface models of the Inventory Visibility (IV) applications. The IV application local interface model is, again after the required transformation into RDFS, first annotated and then reconciled with the eKanban RO. The outcome of these activities is a set of Reconciliation Rules that effectively maps IV application interface schema instances (transformed into RDF instances and in conformance with their model) into and out of the RO elements. In this way, the private IV application interface model (e.g., proprietary XML Schema) is reconciled with the public interface model (i.e., the IV&I eKanban BOD XML schema annotated with the eKanban RO). These reconciliation rules, effectively enabled alignment of semantics of the applications' interfaces with the eKanban message standard using the RO by making references to the RO from both the private and public side of the architecture.
- As a result of the public and private side of modeling and reconciliation, the Reconciliation Rules are made explicit and are based on the model's intended meaning of the document elements rather than on how the document elements are to be encoded and transacted, as is currently the state of practice. Once the Reconciliation Rules are successfully defined, the vendor could execute these rules using the ARES run time engine. The Reconciliation Rules may be reused for the same concepts used across message schemas. Successful reconciliation was accomplished by two independently developed applications: Apolon, capable of sending and receiving only its proprietary version of the AuthorizeKanban message in RDF format and the GM-developed IV application also capable of sending only its own proprietary version of the same message in XML format.

- The GM and Apolon IV applications could use the Gateway to (1) invoke the reconciliation function using the semantic mediation engine; (2) call AXTOR and/or ARTOX service to transform the RDF instance from and/or to an XML instance conformant to the BOD schema; and (3) to send a message over the wire.
- Finally, the Gateway used the Johnson WS Execution Engine that allows Web Services profile specification and a WS call execution.

5. Results

Our implementation of the proposed semantics-based architecture allowed us to demonstrate the resultant semantic middleware capability for an eKanban message exchange between Inventory Visibility (IV) applications and the Inventory Visibility and Interoperability (IV&I) test harness.

At run time, the GM and Apolon applications communicated via their proprietary eKanban messages (i.e., variants of *AuthorizeKanban* message) with the Gateway (shown in Fig. 1 that effectively played the role of the semantic middleware) that, in turn, invoked respective reconciliation rules and enabled creation of the eKanban standard-conformant messages. In this way, the Gateway enabled an interoperable exchange of messages from the sender applications to the recipient applications. The messages were sent “over the wire” by the Gateway that used the ATHENA Johnson Web Services (WS) execution tool.

In our distributed demonstration, the GM application (running in USA) successfully sent an authorization message to the Ford Test Harness (also running in USA) that, in turn, successfully processed the message. In parallel, the GM application sent the same authorization message to the Apolon IV application (running in Serbia). The Apolon application successfully processed the message and, as a consequence, sent an authorization message to FTH. To execute its mediating role, the Gateway invoked the ATHENA reconciliation engine running on a server in Italy.

The semantic middleware capability was demonstrated (1) to be able to handle multiple message formats and WS execution approaches from independent, heterogeneous applications; and (2) to provide required transformations and model-based reconciliations to result in interoperable messages over the wire. Fig. 2 summarizes the transformation and reconciliation capabilities provided by the semantic middleware to achieve interoperable IV applications that maintain proprietary communication solutions. These capabilities are listed below.

- GM legacy enterprise application has proprietary XML Schema-based data exchange interface and used AXIS web services synchronous execution approach [10]. GM communicated its *Authorization* message (in the GM proprietary format) to both FOS and FTH via the middleware.
- The Apolon IV application has a proprietary RDF Schema-based data exchange interface and used SAAJ web services synchronous execution approach [11]. FOS received from GM application the *Authorization* message via the middleware (in the FOS proprietary format) and sent *KanbanAuthorization* message (again in the FOS proprietary format) to the FTH via the middleware.
- The FTH had an IV&I-conformant XML Schema-based interface and used AXIS-based web services asynchronous execution approach implementing WS-I Basic Profile (BP) and the WS-I reliable and secure profile (RSP) [7].
- The ATHENA-enabled Semantic Middleware used a reconciliation engine and transformation tools to transform messages into appropriate representation for the receiver. It received as an input either proprietary XML or RDF Schema-based document instances from either of the two applications that used either AXIS-based or SAAJ-based web services synchronous execution approach. On the output, it generates IV&I-conformant XML Schema-based eKanban messages that were sent

using the Johnson-based web services asynchronous execution approach implementing the WS-I Basic Profile and the WS-I Reliable and Secure Profile (RSP) [7].

In our demonstration, within the Enterprise Modeling area, we focused only on data exchange requirements capture and interoperable exchange of models among EM tools. Ideally, the IV&I Enterprise Model should be closely tied to developing IV&I Reference Ontology in support of the eKanban data exchange. This is indicated with the dotted line connecting the IV&I EM and RO in Fig. 1. Effective integration of these two phases, however, remains a challenging research issue for the future.

	GM / Apolon	ATHENA-enabled Semantic Middleware	Ford Test Harness (FTH)
Function	GM: Legacy Enterprise App. FOS: IV App.	Semantic Middleware	Test Harness
Message Specification	GM: XML Schema FOS: RDF Schema	Input: XML or RDF Schema Output: IV&I BOD (XML Schema)	IV&I BOD (XML Schema)
WS Execution	GM: WS Sync (AXIS) FOS: WS Sync (SAAJ)	WS Asynch BP (RSP) (Johnson+WSS Plugin)	WS Asynch BP (RSP) (AXIS)

Figure 2: Transformation and Reconciliation Capabilities of the ATHENA-enabled Semantic Middleware

6. Business Case

New standards-development methods for interoperable systems, such as the one described in the paper, are needed so that large-scale interoperable systems may be realized efficiently within and across industry verticals. The prototypical architecture and implementation represents a first step in the direction of engaging all the stakeholders – from academia to industry – to take part in developing a realistic, viable platform that can enable effective and efficient large-scale interoperable systems development. Today, this is not possible.

Pursuing innovative platforms for increased efficiencies in support of supply chain designs and implementations are necessary to answer to global competitive pressures that have caused manufacturing industries to aggressively pursue ways to reduce cost and shorten lead-time. The novel architecture may enable automated trading collaborations that work seamlessly to result in improved data accuracy, elimination of wasted manual operations, interactive decision making, and better quality as a result of reducing premium freight and inventory buffers. The results achieved using the proposed semantics-based architecture provide initial evidence that increased automation and greater reliability are possible for future electronic business collaborations in industrial supply chains.

7. Conclusions and Future Work

We have presented our experience in prototyping a novel, semantics-based architecture for standards-based approaches to interoperable applications and in support of large-scale interoperable systems development. Significantly, our demonstration showed that semantic integration technologies begin to form a basis of a "semantic middleware" capability to manage heterogeneous applications' data exchanges by enabling model-based standard messaging interfaces in a flexible, yet precise manner.

8. Acknowledgements

The authors acknowledge the contributions of many collaborators to the ideas presented in this paper: Mohammad Abidi, Nenad Anicic, Ed Barkmeyer, Claudia Guglielmina, Marija Jankovic, Al Jones, Jaewook Kim, Thomas Knothe, Zoran Kokovic, Don Libes, Igor Miletic, Michele Missikoff, Ivana Novicic, Seog-Chan Oh, Klaus-Dieter Platte, Lorenzo Pondrelli, Rainer Ruggaber, Richard Stevens, Eswaran Subrahmanian, Francesco Taglino, Julien Vayssière, Faisal Waris, Jungyub Woo, Shang-tae Yee, and Gary Yu.

9. 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] *ATHENA Integrated Project Web Site*, accessed April 2007. Available at www.athena-ip.org

[2] Ivezic, N., Barkmeyer, E., Kulvatunyou, S., Jones, A., Snack, P., Marjanovic, Z., and Cho, H. *A Validation Architecture for Advanced Interoperability Provisioning.* In Proceedings of Exploiting the Knowledge Economy: Issues, Applications, and Case Studies, pp. 157-164, P. Cunningham and M. Cunningham (Eds.), IOS Press, 2006.

[3] *XML Schema Part 1: Structures Second Edition*. Available at http://www.w3.org/TR/xmlschema-1/

[4] *Automotive Industry Action Group (AIAG) Web Site*, accessed April 2007. Available at http://www.aiag.org/

[5] *Resource Description Framework (RDF) Specification*. Available at http://www.w3.org/RDF/

[6] Miletic, I., Vujasinovic, M., Ivezic, N., and Marjanovic, Z. *Enabling Semantic Mediation for Business Applications: XML-RDF, RDF-XML, and XSD-RDFS Transformations.* In Proceedings of International Conference on Interoperability of Enterprise Software and Applications, pp. 483-494, R. Goncalves, J. Muller, K. Mertins, M. Zelm (Eds.), Springer-Verlag, 2007.

[7] *WS-I Organization Web Site*, accessed April 2007. Available at http://www.ws-i.org/

[8] Novicic, I., Kokovic, Z., Jakovljevic, N., Ljubicic, V., Bacetic, M., Anicic, N., Marjanovic, Z., and Ivezic, N. *A Case Study in Business Application Development Using Open Source and Semantic Web Technologies.* In Proceedings of International Conference on Interoperability of Enterprise Software and Applications, pp. 483-494, R. Goncalves, J. Muller, K. Mertins, M. Zelm (Eds.), Springer-Verlag, 2007.

[9] *Open Applications Group (OAG) Web Site*, accessed April 2007. Available at www.openapplications.org

[10] *Apache AXIS Web Site*, accessed April 2007. Available at http://ws.apache.org/axis/

[11] *Soap with Attachments API for Java (SAAJ) Project Web Site*, accessed April 2007. Available at https://saaj.dev.java.net/