Semantic Mediation Bus™: An Ontology-based Runtime Infrastructure for Service Interoperability

Wen Zhu
Alion Science and Technology
Vienna, VA, USA
wzhu@alionscience.com

Abstract—**In this industry case study, we introduce the concept of semantic mediation that enables service interoperability through common ontologies, even when the services are implemented using different data models and message standards. Leveraging open standards such as Web Ontology Language (OWL) and Semantic Annotations for XML Schema and WSDL (SAWSDL), the Semantic Mediation Bus™ (SMB) removes the need to perform manual correlation among disparate data sources by extending the Enterprise Service Bus (ESB) infrastructure for ontology-based runtime service mediation.**

Keywords-Semantic Web, Service Oriented Architecture, Enterprise Service Bus, Service Service Provisioning, SAWSDL

I. Introduction

The industry recognizes Service Oriented Architecture (SOA) as a key enabler for service interoperability [1]. To achieve interoperability, service producers and consumers need to adhere to a set of common messaging standards. Furthermore, they need to agree on a common data model to describe the business concepts conveyed in message payloads. While this approach may work fine within the boundaries of an enterprise, it is often impractical to expect all the trading partners in a diverse community to agree to a single set of business terminologies, let alone a standard data model for all service implementations.

On the other hand, ontologies are designed to capture and model common business concepts underlying different terminologies used by different organizations. It would seem natural that ontologies could be used to argument the service descriptions in order to overcome traditional SOA's reliance on a common data model. While ontologies have been used for enhanced service discoverability in many organizations, often used in conjunction with a Service Registry, they are still primarily viewed as a design time tool. Developers consult ontologies to understand the business concepts and how they map to data models used by different organizations, then go on to develop the service providers and consumers.

We argue that machine-readable ontologies could be used directly by a runtime middleware infrastructure to perform semantic mediations between service providers and consumers, thus eliminating the need for custom mapping development. We will present such an infrastructure, Semantic Mediation Bus™, in this paper.

This paper is organized as follows. First, background information about SOA and its infrastructures are provided, along with their benefits and limitations. Next, we discuss the motivation for semantic mediation using an example use case. We then introduce the architecture and the reference implementation of the Semantic Mediation Bus™. Finally, we present our conclusions and discuss related works.

II. Background

SOA is an architectural style for building software applications that uses services available within an enterprise or externally on a network, such as the Web [2]. SOA promotes loose coupling between services, allowing business applications to easily re-use existing services and resulting in applications that are easily and quickly built or changed. Key characteristics of SOA include:

- A Service is a generic implementation of well-defined business functionality which can be used and reused by other applications. An important aspect of SOA is the separation of the service interface from its implementation.
- Services are self-contained and loosely coupled.
- Services can be dynamically discovered.
- Services can be accessed using standard communication protocols.
- Composite services can be built by aggregating other services.

Even though we focus on SOA as a set of architectural principals and approaches in this paper, we should note that the scope of SOA is much broader than technology. It helps align IT services with business missions, and is a critical part of enterprise architecture methodologies.

A key component of an enterprise's SOA infrastructure is the Enterprise Service Bus (ESB). ESB is a flexible connectivity infrastructure for integrating applications and services [3]. An ESB powers a SOA by reducing the number, size, and complexity of interfaces between those applications and services.

One of the ESB's responsibilities is to perform protocol adoption. With an ESB serving as the integration hub and so long as the service consumer understands the data model used by a service provider, the service can be access using a different protocol than that originally used by the provider. Many ESBs are also capable of simple message transformations based on technologies such as the Extensible Stylesheet Language Transformations (XSLT) [4]. In many cases, this may be sufficient to allow a few consumers to access one service provider. The maintain burden grows exponentially once additional service providers for the same service, as well as potentially different data models used by these providers are introduced. Manual intervention and deep domain knowledge is required to develop custom mappings to correctly use data exposed by these related but different services.

III. Semantic Mediation Bus™: Concept and Implementation

A. Motivation

We believe the existing ESB infrastructure can be extended to enable services built with different data models to interoperate, so long as these data models describe the same set of business concepts and the their relationship among these models have been clearly defined in a machine-readable format. We refer this additional ESB capability as Semantic Mediation, as oppose to traditional protocol adaption and message transformation.

The motivations for Semantic Mediation include:

- Support for on demand integration of data sources. There are cases where the particular service provider from which a client designed to consume data is not available. In these cases, the client needs to be able dynamically discover services with similar capabilities and be prepared to consume data from those services, even though the format in which the data is provided may be different from the format the client was designed to consume.
- Streamline service integration process. When organizations make the investment in ontology development, the ontologies are mainly viewed as a knowledge capture and analysis tool. It often takes costly development efforts to translate that knowledge to executable artifacts. By leveraging semantic mediation, organizations can reduce or eliminate the development cost.
- Reuse existing SOA infrastructure. As organizations embrace semantic technologies, it is impractical to expect them to stand up an entirely different set of technology components after significant investment in SOA infrastructures. Thus, any semantic interoperability components need to be seamlessly embedded in and work with the existing SOA infrastructures to reduce the total cost of ownership.

To illustrate the need for a semantic mediation infrastructure, consider the following scenario: a flight track display application is designed to consume data from a Flight Track Web Service offered by the US Federal Aviation Administration (FAA). When the need to integrate with an Air Force Flight Tracking Web Service arises, the client developer needs to first understand the Air Force service data format and message syntax by talking to a Subject Matter Expert, even though the second service provides essentially the same information. Then custom mappings need to be developed to allow the client to consume the Air Force data. Finally, the newly developed mappings need to be integrated to the flight track display application. This process needs to be repeated for each additional service provider the display application interacting with.

B. Concept Overview

Semantics Mediation Bus™ (SMB) is a runtime infrastructure we have developed to enable semantic interoperability through common ontologies, even if the services are implemented using different data models and message standards.

Cooperation through federation, instead of standardization is the principle behind using ontology as the tool to describe message meanings. This ontology driven approach avoids imposing a standard that has to be agreed by everybody, which is especially important when information needs to be shared across organizational boundaries. The SMB allow each organization to select the standards best suited for their business needs, while still able to use services offered by the larger community.

Furthermore, we have identified the following design principles for the SMB:

- Leverage SOA infrastructure, instead of replacing it. The SMB needs to integrate with existing SOA infrastructure components such as ESB and Service Registry and delegate identified functions such as protocol adaption, message transformation, and security management to these components.
- Focus on an extensible framework, instead of more features initially. Our effort is first and foremost aimed at identifying the building blocks – technology standards, functional requirements, and reusable components – required for semantic mediation.
- Adhere to open standards, instead of proprietary solutions. The SMB needs to be adaptable to existing SOA infrastructures that an organization may have, and modular so that it can be combined with other products to create best-of-breed solutions.

C. Technology Foundation

Open standards play an important role in SMB. Figure 1 illustrates some of the key standards that enables runtime mediation.

Figure 1 Technology Foundation

Built on an SOA infrastructure and ESB in particular, SMB assumes standards such as the Extensible Markup Language (XML) [5] and the Universal Resource Identifier (URI) [6] are used to encode the actual messages. This implies that SMB will not directly operate on unstructured data, like most SOA infrastructure components.

Furthermore, we assume the services are described and exposed via such standards and practices as Web Service Description Language (WSDL) [7], SOAP [8], and Representational State Transfer (REST) [9].

Among them, WSDL is particularly important for service discovery, as it provides the mechanism for service interface information including the syntax of the payload to be specified. However, WSDL does not provide a way to describe the meaning or the semantics of the data or the behavior, required if the infrastructure is to have the intelligence to bridge different data models used by different service providers. To accomplish that, we need a set of standards that connect services and data to their meanings captured in ontologies. The SMB uses following standards as the foundation for semantic mediation:

- Web Ontology Language (OWL) [10]. For our purpose, ontologies support exchange and reuse of data and information through the formulation of precise and shareable descriptions of a given domain enables complex description of data and data relationships, translation of vocabulary from one community to another and ready access to semantic tools to fully enable discoverability of data and services across the enterprise.
- Semantic Annotation for WSDL and XML Schema (SAWSDL) [11]. SAWSDL defines mechanisms using which semantic annotations can be added to WSDL components. SAWSDL does not specify a language for representing the semantic models, e.g. ontologies. Instead, it provides mechanisms by which concepts from the semantic models that are defined either within or outside the WSDL document can be referenced from within WSDL components as annotations. These semantics when expressed in formal languages can help disambiguate the description of Web services during automatic discovery and composition of the Web services.
- Lightweight Semantic Descriptions for Services on the Web (WSMO-Lite) [12]. WSMO-Lite authors regard it as the next evolutionary step after SAWSDL, filling the SAWSDL annotations with concrete semantic service descriptions. The ultimate goal is to support real-world challenges in intelligent service integration.

D. *Semantic Service Provisioning*

At a high level, runtime mediation requires the SMB to have the intelligence to:

1. Assess the compatibility level between two services, even when these services have data models and message formats;
2. Once the compatibility between two services is established, bridge the data models difference by performing transformations.

Thus, SMB requires services to be semantically provisioned, i.e. the elements of the service interface description, specified in WSDL and XML schema, be mapped to element in ontologies to give them "meaning". Semantic service provisioning is illustrated in Figure 2.

Semantic provisioning associates the elements of the service and its payloads, described in traditional artifacts, to common concepts captured in ontologies. More specifically, elements of service artifacts are mapped to two different kinds of ontologies:

- An ontology about the business concepts carried in the message payloads, which we refer to as an Enterprise Vocabulary. Using the modelReference SAWSDL annotation, we can associate the XML elements in the payload, which can differ from service provider to service provider, to elements in the Enterprise Vocabulary. In this example, we assert that the FlightTrack XML type correspond to the AirTrack concept.
- Ontologies describing the services themselves based on standards such as the Minimal Service Model (MSM) [13]. We refer to these ontologies as Service Ontologies. Similarly, we can associate actual web service instances to concepts in the Service Ontology.

For the purpose of bridging the data model difference, the Enterprise Vocabulary can serve as an intermediary data format. For example, transformation rules can be defined

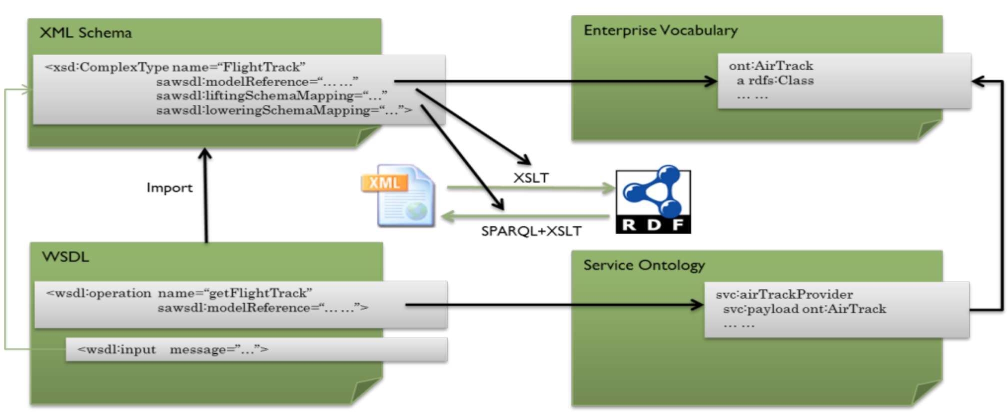

Figure 2: Semantic Service Provisioning

between business concepts represented using RDF/XML and the actual XML types used by the services. Using SAWSDL, the following rules can be defined:

- Lifting Rules – XSLT can be used to specify how an XML payload element can be transformed to the RDF/XML representation, allowing the SMB to use the ontology as the intermediary data format.
- Lowering Rules – To transform the payload from the intermediary ontology format back to an XML representation, the SMB uses a SPARQL query to extract key pieces of information the intermediary ontology and serialize it back to XML using XSLT.

In the simplest case, two service providers have mapped their services and payloads to the same set of ontologies, i.e. there is a common Enterprise Vocabulary and a common Service Ontology. In a more complex scenario, each service provider may have done their ontology development independently. For the later cases, mapping between the two set of ontologies needs to be perform using standard ontology tools.

We should note that in our SMB implementation, SAWSDL is used to annotate the WSDL and XML schema. However, it is conceivable that other means could be used instead. For example, Service Oriented Architecture Modeling Language (SoaML) [14] could be used to describe the services and annotations could be applied to the model elements using UML stereotypes.

E. Semantic Mediation Bus Architecture

As shown in Figure 3, we design the SMB to be a thin layer embedded in an ESB. At the core of the SMB is a Semantic Mediator that performs the following functions:

- Message Schema Mapping, bridging the message format difference between the service consumer and the service provider. The message transformation rules are predefined and available in the Service Registry.
- Web Service Proxy, exposing a web service endpoint on the ESB based on the syntax and the meaning of the messages that the consumer will accept.
- Semantic Interoperability Assessment, assessing the interoperability level between the service for which the client is designed for and the service that is available.

Though it is possible for the SMB to dynamically discover an alternative service provider based on the service the consumer is designed to consume and automatically route the consumer's request to that provider, a human in the loop scenario is more likely given the current state of technologies. In our SMB design, it is the service consumer's responsibility to specify both

- The service for which the consumer is designed for, expressed in WSDL, and
- The alternative service to invoke.

The Semantic Mediator is plugged into an ESB through an ESB Adapter using the ESB's Application Programming Interfaces (API), such as the Java Business Integration (JBI) API [15][16].

The SMB also features an Extension Framework that allows different parts of the SMB to be customized based on business needs. For example, there may be need to specify message transformation rules in a way other than using SAWSDL lifting and lowing rules. Another extension point is the interoperability assessment algorithm. For example, in the case where multiple service providers have developed different ontologies for their services, the interoperability assessment will likely involve the use ontology matching algorithm such as that described in [17].

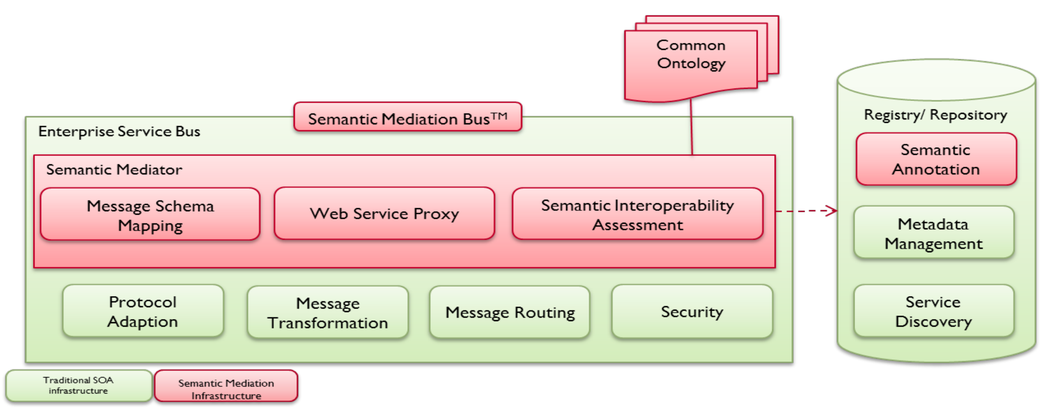

Figure 3: Semantic Mediation Bus Architecture

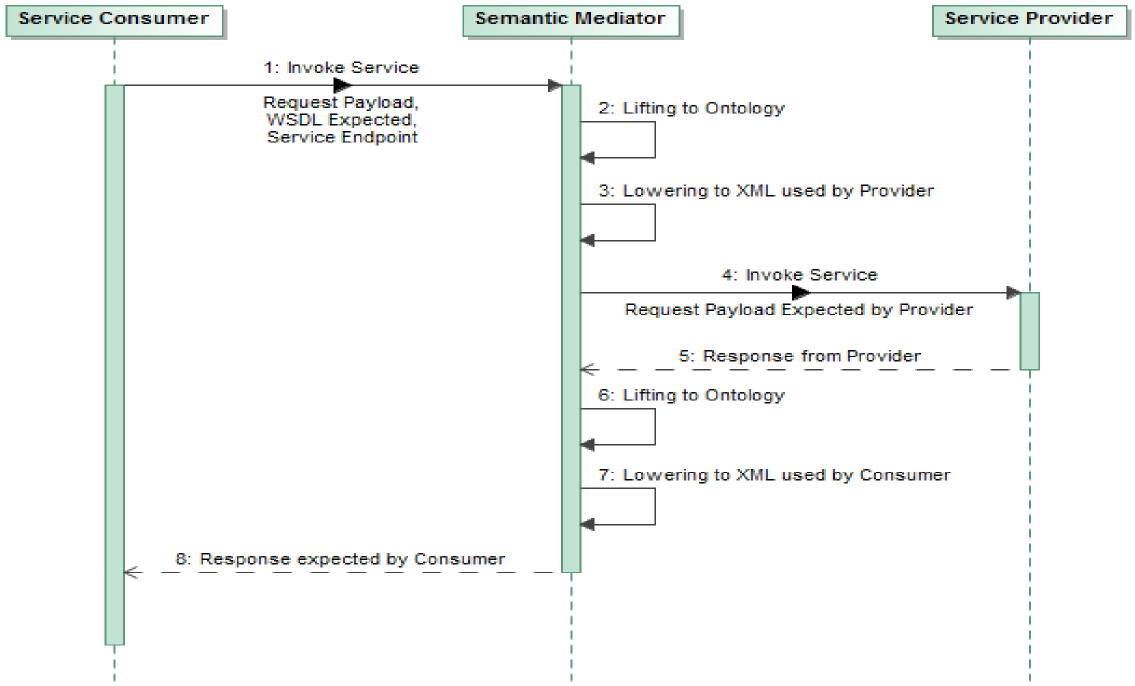

Figure 4: Message Transformation

F. SMB Reference Implementation

Our SMB reference implementation uses Apache Synapsis ESB. It is deployed in the Amazon Elastic Computing Cloud (EC2) and available at http://www.alionsoa.com.

This implementation uses SAWSDL for lifting and lowing rules. The interoperability assessment component in this implementation has limited functionality and requires exact match between two services for semantic mediation, that is:

- The service definitions, as specified in the WSDL, references the same concept in the Service Ontology;
- The XML types for the message payloads references the same concept in the Enterprise Vocabulary.

Figure 4 shows how the SMB performs message transformation at runtime:

- The SMB is invoked by a service client who provides a pointer to the WSDL of the service it is designed to consume (the original provider). It may also specify the service it is looking to use (the alternate provider).
- The request arrives at a generic end point exposed by the SMB. The SMB does not pre-generate web service endpoints for the services it mediates. Rather, service information required for mediation will be supplied by the client at runtime as parameters.
- Uses the annotations contained in the WSDL for original and alternate providers, the SMB locates and executes transformations rules to first lift the client request to an immediate format expressed in RDF/XML and then lower it to the message format expected by the alternate provider.
- The SMB invokes the alternate provider.
- For the response message, it uses similar lifting and lowering rules to transform the alternate provider's response to a format understood by the client.

IV. Related Work

Semantic provisioning of services is the requisite for semantic mediation. For that, we have developed a Semantic Metadata Catalog and Portal.

Unlike a traditional web service registry, the Semantic Metadata Catalog and Portal only publishes the syntax of the service interfaces, but also describes the meanings of the messages and their relationships, service discoverability and understandability is greatly enhanced. For example, potential consumers can now discover services based on more than just simple key word search in the registry. The infrastructure will have the intelligence to understand the “meaning” behind the search term and present the relevant information. It allows Communities of Interest (COI) to enrich service descriptions by providing additional metadata. Examples of such metadata include:

- Associations linking a service to the enterprise architecture artifacts describing its business context
- Relationships among services

- Annotations linking service payloads to business concepts captured in ontologies
- Designation of Authoritative Data Sources (ADS)

V. Conclusion

Ontologies can do more than just being design time artifacts. To fully realize the value of semantic technologies, we believe technical capabilities should be developed to demonstrate how investments in ontology development integrate with and provide immediate benefit an enterprise's existing IT infrastructure.

The SMB is such an example. By leverage ontologies at runtime for semantic mediation, we can streamline service integration process by shortening development lifecycle and eliminating the need for custom message mapping, while at the same preserving organizations' investment in existing SOA infrastructures.

Acknowledgment

Our work could not have been possible without the support of Alion's leadership team. In particular, the author wants to thank Erik King, Paul Sautter, Carlynn Thompson, and Pedro Cadenas, Jr. and for their support and guidance; and thank Sumeet Vij for his leadership in the development of the SMB. We also want to acknowledge Kyle Loomis, Alexandro MayaBertin and Lowell Vizenor for their outstanding contributions. Last but not least, we appreciate the comments from the reviewers, which helped improve the paper greatly.

References

[1] Alex Cullen et al. "The Enterprise Architecture of SOA", Forrester, 2006.

[2] S Jones, "Toward an Acceptable Definition of Service", IEEE Software, vol. 22, no. 3, pp. 87-93, May/June 2005.

[3] M. Keen et al. "Patterns: Implementing an SOA Using an Enterprise Service Bus", IBM, July 2004

[4] XSL Transformations (XSLT) Version 1.0, W3C Recommendation, November 1999, http://www.w3.org/TR/xslt

[5] Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation, November 2008, http://www.w3.org/TR/xml/

[6] T. Berners-Lee et al. "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, Internet Engineering Task Force, January 2005

[7] Web Services Description Language (WSDL) Version 2.0 Part 0: Primer, W3C Recommendation 26 June 2007, http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626

[8] SOAP Version 1.2 Part 0: Primer (Second Edition) W3C Recommendation 27 April 2007, http://www.w3.org/TR/2007/RECsoap12-part0-20070427/

[9] Roy T. Fielding, "Representational State Transfer: An Architectural Style for Distributed Hypermedia Interaction", 2000, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[10] OWL Web Ontology Language Overview, W3C Recommendation, February 2004, http://www.w3.org/TR/owl-features/

[11] Semantic Annotations for WSDL and XML Schema, W3C Recommendation, August 2007, http://www.w3.org/TR/sawsdl/

[12] WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web, W3C Member Submission, August 2010, http://www.w3.org/Submission/WSMO-Lite/

[13] Semantic Technology Institute International Working Group: "Minimal Service Model", http://cms-wg.sti2.org/minimal-service-model/

[14] Service Oriented Architecture Modeling Language (SoaML), Version 1.0, Object Management Group (OMG), March 2012, http://www.omg.org/spec/SoaML/1.0/PDF

[15] Java Business Integration (JBI) 1.0, JCP, August 2005, http://jcp.org/en/jsr/detail?id=208

[16] Wen Zhu, Walt Melo, "Refactoring J2EE Application for JBI-Based ESB: A Case Study", pp.213-217, 2009 IEEE International Enterprise Distributed Object Computing Conference , 2009

[17] Kelly Mora et al: CompositeMatch: Detecting N-ary Matches in Ontology Alignment, Proceedings of The Fourth International Workshop on Ontology Matching, Washington DC., USA, October 2009