WSMX – An Architecture for Semantic Web Service Discovery, Mediation and Invocation

Matthew Moran, Adrian Mocan
Digital Enterprise Research Institute (DERI),
National University of Ireland, Galway
matthew.moran@deri.org, adrian.mocan@deri.org

Abstract

This paper introduces an implementation of an event driven service oriented architecture for the execution of Semantic Web Services based on the emerging Web Service Modeling Ontology, WSMO [Roman et al., 2004].

1 Introduction

Web Services constitute one of the hot areas in Web technology supporting the remote invocation of business functionality over the Internet through message exchange. Despite the current proliferation of Web Services, most of those used today are only one-way data retrieval or update functions. While useful, they do not provide the critical components required by B2B systems. The complex issues surrounding B2B integration are well documented but no Semantic Web Service software infrastructure currently exists that tackles this problem. Building on the most recent research into Semantic Web Services, the Web Services Execution Environment, WSMX, aims to provide such a solution.

WSMX, allows Web Services whose semantics have been formally described to be discovered, selected, mediated and invoked to carry out specific client tasks. Semantics, in this context, is the meaning of various aspects of Web Services that allow machines to automatically carry out tasks using Web Services with a minimum or no human intervention. The conceptual model used by WSMX, WSMX-O [Cimpian et al., 2004], has its foundation in the Web Service Modeling Ontology (WSMO) and the Web Service Modelling Framework, WSMF, [Fensel & Bussler, 2002]. The goal of WSMX is to provide a flexible environment for application and business integration based on strongly decoupled physical components with strong mediation services enabling every party to speak with each other as advocated in WSMF. The Web Services Modelling Ontology, WSMO, is a Semantic Web Services initiative being led by DERI and several EU projects to provide an ontology describing various aspects of Semantic Web Services. WSMX is a reference implementation providing an event-based service oriented architecmented for the first version of WSMX. The WSMO Editor is used to create WSMO descriptions of Web Services, ontologies, mediators and goals. WSMX provides a WSDL interface to accept these descriptions ture to manage the tasks of discovery, mediation, choreography, orchestration and invocation associated with the execution of a business goal. Internally the execution semantics of WSMX will provide the reliability, security and trust required of an industrial B2B integration system.

2 Architecture and Implementation

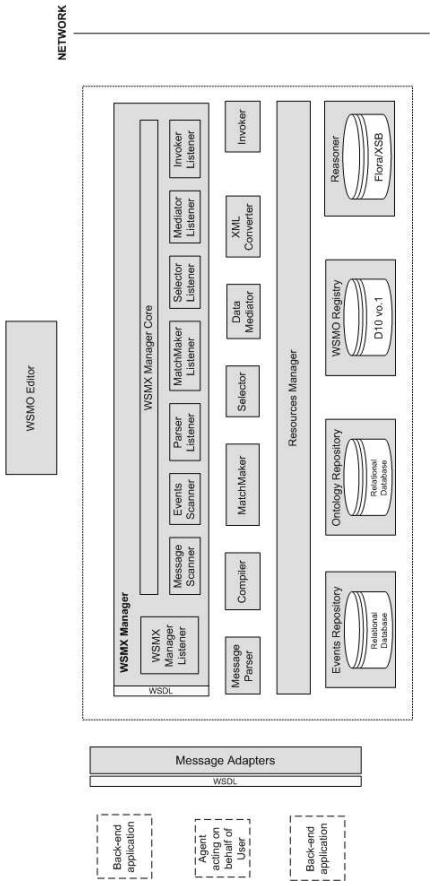

Figure 1 WSMX Architecture

Figure 1 shows the architecture that has been imple

and direct them to the Compiler component. Compilation means validating the descriptions and storing them persistently in the Ontology Repository.

The WSMX Manager controls the operational flow of the system and is also responsible for event management, keeping a full record of the life cycle of each of the various data events created as a user goal request is processed by the system. WSMX provides a Web Service interface described in WSDL to accept service requester goals. The WSMX Manager regularly scans for new messages. Once a new message representing a requester goal is picked up, it is decomposed, validated and translated into an internal persistent WSMX representation by the Message Parser. The Matchmaker then attempts to match the goal to capabilities of Web Services known to WSMX. The concept of capability is used in WSMX-O to formally describe the functionality a Web Service offers. Any Web Service with a capability that matches the goal is returned by the Matchmaker. Matching services are found by comparing the logical expressions representing the postconditions and effects of the requester goal with those of the Web Service capabilites. The Selector component selects the Web Service that provides the best match for the goal based on service requester preferences. The Data Mediator mediates the data provided by the service requester to the ontology defined by the service provider. Mediation is described in more detail in the next section.

Once the data has been mediated, it may need to be translated from a logical language format to an XML format before it is included in the SOAP message sent by the Invoker to the target service. This translation is the job of the XML Converter. Once the data is in a format that can be used during Web Service invocation, the Invoker makes the actual Web Service invocation on the selected Web Service using the mediated and converted data.

3 Mediation

WSMX does not assume that all Web Services share one conceptualization of the world. Rather it takes the view that the applications represented by Web Service interfaces are normally heterogeneous and autonomous. This gives rise to the need for mediation at data, process and business protocol levels. This version of WSMX starts with the problem of data mediation.

The mediation component of WSMX is designed as a service and provides mapping rules from a source ontology (usually the one used by the service requester) to the target ontology (the one used by the service provider). Each mapping rule contains a set of links between concepts (attributes) from the source and concepts (attributes) from the target ontology. These links specify the operations that have to be applied when the rule is executed and the required target instances are created. The mapping rules are created semi-automatically in order to reduce the amount of work required from the user side as much as possible and to offer consistent assistance when necessary. By using a graphical user interface the user can view the internal structure of concepts, may use the mapping suggestion offered and then create the desired mappings. The suggestions are made based on the concepts structure and on mappings that already exist. Figure 2 shows the relation between the mediator component and the execution environment.

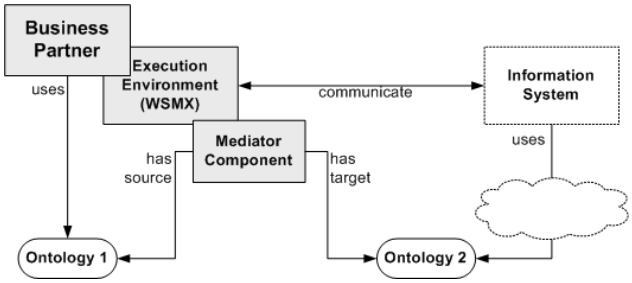

Figure 2 WSMX Architecture

The Execution Environment communicates with a Business Partner and an Information System each using a different ontology. When a set of instances have to be passed from the Business Partner to the Information System the Mediator Component is called to perform the transformation from source to the target format.

4 Next Steps

The current implementation of WSMX provides a minimum but meaningful framework for executing Semantic Web Services. The next phase of our approach will concentrate on refining the components described in section 2 and extending the existing architecture to accommodate the concepts in WSMO Standard such as those for Web Service choreography and orchestration. In particular we will concentrate our attention on extending the design of existing components for Semantic Web Service discovery, selection and invocation. The Mediation component will also be enriched by adding process mediation support.

Acknowledgments

The work is funded by Science Foundation Ireland under the DERI-Lion project.

References

[Cimpian *et al.*, 2004] Emilia Cimpian, Adrian Mocan, Matthew Moran, Eyal Oren, Michal Zaremba. WSMX Conceptual Model. v.0.1, available at http://www.wsmo.org/2004/d13/d13.1/v0.1

[Fensel & Bussler, 2002] Dieter Fensel and Christoph Bussler, The Web Service Modeling Framework WSMF. Electronic Commerce Research and Applications, Vol. 1, Issue 2, Elsevier Science B.V.

[Roman *et al.*, 2004] Dumitru Roman, Holger Lausen, Uwe Keller. The Web Service Modeling Ontology - Standard (WSMO-Standard) v. 0.2 available at http://www.wsmo.org/2004/d2/v0.2