Enabling execution of Semantic Web Services - WSMX core platform

Michal Zaremba, Matthew Moran

Digital Enterprise Research Institute, DERI, Ireland
{michal.zaremba, matthew.moran}@deri.org

Abstract. In this paper we present a demo implementation of WSMX – a software system based on WSMF [1] and WSMO [2] – which is a core infrastructure enabling execution of Semantic Web Services, Our demo implementation is based on research carried out by the WSMX[1] working group. The current system implementation is an Event and Service Oriented Architecture (SOA) enabling discovery, selection, mediation and invocation of formally described Web Services to carry out specific client tasks. Although our first implementation of the system is incomplete in terms of the required functionality, all components defined through WSMX working group are in place and the system is able to execute simple client goals expressed in WSMO.

1. INTRODUCTION

Web Services support the remote invocation of business functionality over the Internet through exchange of well defined standardised messages. Existing Web Services' cornerstone technologies such as UDDI [3], WSDL [4] and SOAP [5] provide the basic functionality for discovering required functionality (UDDI), describing interfaces (WSDL) and exchanging well-formed messages (SOAP) in ***heterogonous***, ***autonomous*** and ***distributed*** systems. Despite their potential, Web Services remain almost nothing more than enhanced RPC calls over the Internet. Most available Web Services are only one-way data retrieval and update functions hard coded into client's software. While still useful, they do not provide any automated fulfilment of customers' goals. Web Services cannot be automatically bound to requestor goal at run time, but only at design time of the requestor system, because existing Web Services are not semantically described. We are building our system on the most recent research in the Semantic Web Services. WSMX aims to support basic functionality of B2B & EAI integration servers while fulfilling the promise of automatic (semi-automatic) service discovery, selection, mediation and execution.

[1] Web Services Modeling Execution (WSMX) working group – http://www.wsmx.org/

2. WSMX Core System

The WSMX platform aims to offer a complete support for interacting with Semantic Web Services. These Web Services must provide a WSMO-compliant (WSML [6]) description of themselves specifying their capability, how to interact with them (interfaces) and ontologies they use, along with other non functional properties. Figure 1 presents the architecture and components (based on [7]) that has been addressed in the first version of the system (except discovery).

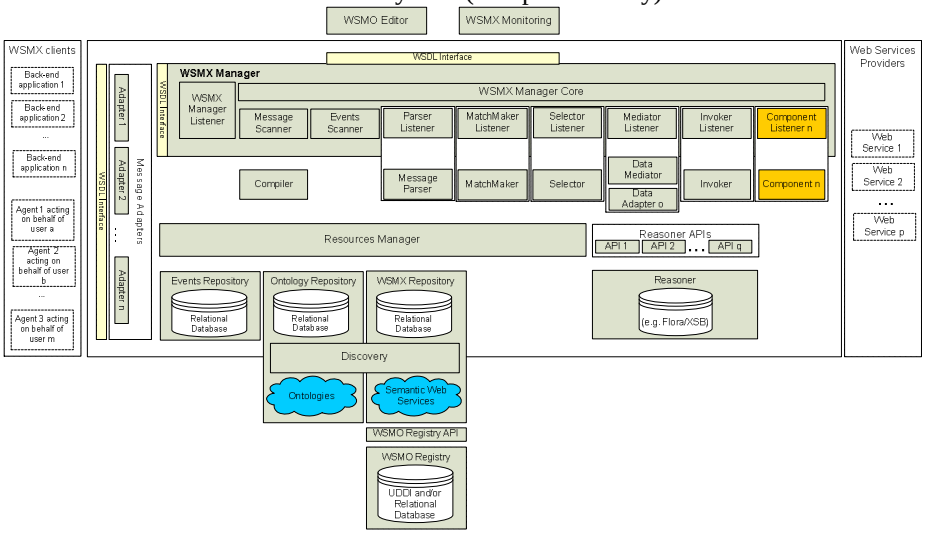

Fig. 1. In WSMX once an element described in WSML has been compiled, it becomes available for WSMX to use during the execution. The WSMX platform can match the semantic descriptions of Web Services' capabilities with semantic descriptions of a goals provided by users. If match can be made, the data mediator provides support for transformation between ontologies of requestors and providers of Web Services. Finally the invoker component makes the call to the appropriate Web Service.

Separately to core WSMX platform, many others supportive components such as editors, monitoring tools or adapters are developed along the way. WSMX provides a WSDL [4] interface to communicate with these additional components.

Event-based communication in the WSMX platform is managed by the event manager. Most of the components never communicate directly (one component never calls the other component) but they create and consume events (or change status of existing events), which are broadcast to all components subscribed to listen for events in the system. Such an approach enables decoupling communication from processing, which enables both flexible communication patterns and a flexible run-time architecture.

3. CONCLUSION AND FUTURE WORK

Web Services merely support the remote invocation of business functionality over the Internet. WSMX - the reference implementation for Semantic Web Services goes much further, as the system aims to enable any information systems to interact with any other information system and to preserve message, process and protocol semantics. Along the way through the open source development[2] we aim to enhance WSMX with intended functionality for all the conceptually defined components. Our next steps will introduce business process engine, dynamic execution semantic engine (future research in [8]), process and protocol mediators and many other core and supportive components, which will enhance WSMX platform.

ACKNOWLEDGMENTS

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

REFERENCES

1. D. Fensel, C. Bussler. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, Vol. 1, Issue 2, Elsevier Science B.V.
2. D. Roman, H. Lausen, and U. Keller. Web Services Modeling Ontology Standard. WSMO Working Draft v02, 2004.
3. UDDI. OASIS specification for Universal Discovery Description and Integration
4. Web Services Description Language (WSDL) 1.2, W3C Working Draft, 3 March 2003
5. SOAP version 1.2, W3C Recommendation, 24 June 2003
6. J. Bruijn, D. Foxvog, E. Oren, D. Fensel. WSML-Core, Working Draft v01, 2004,
7. M. Zaremba, M. Moran. WSMX Architecture. WSMO Working Draft v01, 2004.
8. E. Oren. WSMX Execution Semantics. WSMO Working Draft v01, 2004.

[2] Sourceforge – WSMX hosting site - http://sourceforge.net/projects/wsmx