Integration and Interoperability on Service Oriented Architectures using Semantics

Jose María Álvarez[1] and Antonio Campos López[1]

Fundación CTIC, Gijón, Asturias, Spain,
{josem.alvarez,antonio.campos}@fundacionctic.org,
WWW home page: http://www.fundacionctic.org

Abstract. The dynamism (need of flexibility to changes) of business processes adds an extra difficulty to the maintainability of enterprise applications. This matter implies that applications may not react in time to cover the new client requirements or the new specifications of third parties services. From a temporal point of view, the most relevant factor is the complexity and size of the changes, if an application is well defined and deployed on a flexible architecture the time on development will be reduced. On the other hand, if the application is monolithic, the changes are heavier than at previous approaches. Business processes based on services described using semantics with specific responsabilities can improve the maintainability and evolution of the applications. The present article introduces a proposal to improve the interoperability and integration capabilities in service oriented architectures (hereafter SOA) using semantic technologies. Firstly, we review the keypoints and the different approaches to deploy SOA and business processes. Secondly, we present our approach trying to solve common problems in SOA. A software component architecture for services using semantics is proposed. It is focused on two points: 1) a framework capable of generating BPEL processes automatically using the semantic descriptions of available services and, 2) an execution platform for performing business processes on Enterprise Service Buses (hereafter ESB).

1 Introduction

The key is flexibility [6]. For all major companies[1][2] flexibility is the keypoint to enable added value to their applications. The new environment execution for enterprise applications and business processes is the web, the largest distributed system. Currently, companies are focusing their efforts on the use of services provided by third parties to build composite applications. They also offer their software as a service creating a large set of available services in the web and

[1] Gartner Report: Justifying Service-Oriented Architecture Initiatives, 13 April 2007. ID Number: G00147984.

[2] Gartner Report: Service-Oriented Architecture Overview and Guide to SOA Research, 3 January 2008. ID Number: G00154463.

changing the vision of software development. The implementation of new services based on existing ones helps systems to remain scalable and flexible while growing: SOA. It comprises three major elements: services (e.g WSDL+SOAP or REST), an infrastructure called the enterprise services bus (ESB [1]) and, policies and processes.

The final services provided by companies are usually a collection of coordinated invocations to different services in order to produce a result in the own organization or among others. This coordination is not easy and we need a language as BPEL [3] for specifying interactions with services. Although companies can create their own business processes using BPEL, it is not possible to directly integrate and operate with all available services in the web. Developers need to know how to perform certain tasks to use the services: discover, select or invoke services (manual maintenance). Semantic Web technologies arise to automate these tasks. They should enable to discover, select, compose, orchestrate, invocate and monitor web-based services automatically. To make use of a web service, a software agent needs a computer-interpretable description of the service, and the means by which it is accessed. The objective of semantic web languages as RDF, RDFS or OWL is to establish a common framework within which these descriptions are made and shared. The use of ontologies to describe and declare services provide a compatible representation language to do this.The combination of semantics and current web services orchestrated using BPEL can create a new flexible paradigm for software development.

This paper is structured as follows. Our main contributions are highlighted in the next subsection. In Section 3 previous related work is reviewed. Finally, last section provides the main conclusions and future work.

1.1 Main Contributions

In this paper, the authors propose a framework capable of generating BPEL processes automatically using the semantic descriptions of available services. We also propose an execution platform based on stable tools to deploy and execute the BPEL processes. Our approach should be close to a production environment so we use industrial and stable standards for services (WSDL+SOAP) and semantics (OWL and RDF). The execution environment is provided by products from vendors as JBOSS or Apache. We focus our efforts on aforementioned contributions and not in the creation of new languages or execution environments.

2 Related Work

There are two main initiatives to combine services and semantics:

1. Web Service Modeling Ontology (WSMO [2]) provides ontological specifications for the core elements of semantic web services. It is a meta-model (MOF [3] is used to specify this model) for semantic web services related aspects. It

[3] Meta Object Facility Specification-http://www.omg.org/mof/

refers to the concepts it defines as "elements": *Web Services*, *Goals*, *Mediators* and *Ontologies*. They are described using a language called WSML. On the other hand, WSMX [9] is the execution environment for business application integration where enhanced web services are integrated for various business applications. It is also a reference architecture of OASIS Semantic Execution Environment (SEE) Technical Committee[4]. Although this approach is technically and conceptually valid and it deals with a complete solution for SOA using semantics, their implementations WSMX and IRSIII [8] are still under development in several projects: TRIPCOM, SWING or DIP.

2. Semantic Markup for Web Services (OWL-S [7]) is an ontology to provide three essential types of knowledge about a service: *Service Profile* tells "what the service does", *Service Grounding* specifies the details of "how an agent can access a service" and *Service Model* tells a client " how to use the service", by detailing the semantic content of requests, and the conditions under which the service is executed. This approach is so practical but it does not cover all requirements of SOA.

Neither of these initiatives cover all requirements of SOA in a production environment. However, using BPEL generated from semantic information and an industrial execution infrastructure (ESB) we can fulfill SOA requirements in a production environment.

Besides, there are approaches to add semantic annotations in the service descriptions. In that case the main proposal is the Semantic Annotations for WSDL and XML Schema (SAWSDL [4]). It is a W3C Recommendation 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. In that case, we think that semantic annotations are not enough to cover SOA requirements. This recommendation is the predecessor of WSDL-S. Also, in SUPER project [5] a semantic extension of BPEL has been implemented. They have added semantics to BPEL with a set of elements in the XML Document of BPEL and they have extended the Apache ODE [6] (BPEL Engine) to process these semantic extensions. In our case, we do not need to extend BPEL and we build the solution over stable industrial products.

3 Improving Interoperability and Integration Capabilities in SOA using Semantics

Our approach to design and implement a SOA architecture is based on a set of decoupling components exposed as services (WSDL), communicated under a common protocol (SOAP) and with a sharing knowledge (ontologies). We can briefly summarize the behavior for each component, as Figure 1 illustrates.

[4] http:// www.oasis-open.org/committees/semantic-ex/
[5] http://www.ip-super.org/
[6] http:// ode.apache.org/

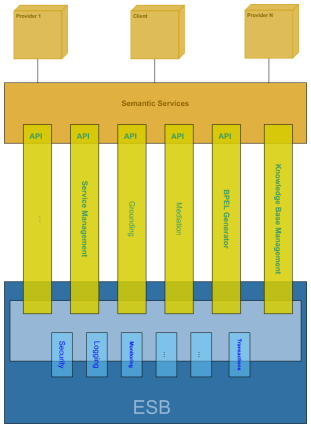

Fig. 1. Architecture for providing Services based on Semantics

- ESB is the execution platform. It is a stable product with a BPEL engine. It could be provided by different vendors: JBOSS SOA, ORACLE SOA+BEA, Apache ServiceMix, OpenESB, Mule Source or Apache ODE. It is the component in charge of BPEL execution.
- Grounding, more specifically data grounding [5], is an infrastructure service. It is the bidirectional process that downgrades a semantic model to a syntactic level through a subprocess called *lowering* and upgrades a syntactic model to a semantic level through a subprocess called *lifting*, enabling actual invocation of web services in SWS environments.
- Mediation, more specifically data mediation, is a service to transform a data model to others. It is a service complementary to grounding. Usually this service is implemented in the ESBs as XSL transformations.
- Service Management is the component in charge of providing store and access to all needed information about services: WSDL, annotations, communication protocols, etc.
- Knowledge Base Management is the component in charge of providing store and access to all needed information about business domain. In our approach, the knowledge base is using ontologies in OWL as knowledge representation and RDF as common data model.
- BPEL Generator is a component that takes the information about services, business domain, a template for a certain business service (created by designers and business dpto.) and a set of production rules [7] to generate a BPEL document implementing the business service. This component generates the BPEL document

[7] http://www.ilog.com/products/ businessrules/

4 Conclusions and Future Work

Our proposal and ongoing research for a SOA architecture using semantics relies on a set of components deployed on an ESB and a BPEL code generator. We are also aware of the intrinsic difficulty of SOA paradigm and that is why we are reusing the previous work in this field. Our design does not differ so much from WSMX but we remark two main different points: 1) We delegate the execution of the business processes in the ESB (stable product). 2) We had previously created the business process. It is created and deployed in the ESB only if a new business service or maintenance tasks are requested. Thus, we gain in robustness and performance on runtime. Our approach will improve the interoperability and integration capabilities of applications in enterprise systems because the added value relies on the use of semantics with a stable execution platform (ESB+BPEL), the gamble of the main vendors in enterprise applications and services as Oracle, JBOSS or Apache for these technologies is the best support to our approach.

On the other hand, our solution is only focused on WSDL+SOAP services but we are working to extend our solution to support other kinds of service (message protocol and description format), such as REST services, WSDL 2.0, etc. We also are working on the first version of the implementation. We would like to align of our solution with other proposals and recommendations from W3C, OASIS and OMG.

Acknowledgements

This work is part of PRIMA project[8], partially funded by the Spanish Ministry of Industry, Tourism and Commerce, cod. TSI-020302-2008-032, and leaded by Teléfonica I+D.

References

1. David A. Chappell. *Enterprise Service Bus.* 2004.
2. Jos de Bruijn, Christoph Bussler, John Domingue, Dieter Fensel, et al. Web Service Modeling Ontology (WSMO). W3C member submission, W3C, 2005. http://www.w3.org/Submission/WSMO/.
3. IBM Diane Jordan and Microsoft John Evdemon. Web Services Business Process Execution Language Version 2.0. OASIS standard, OASIS, 2007. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.
4. Joel Farrell and Holger Lausen. Semantic Annotations for Wsdl and Xml schema. W3C recommendation, W3C, 2007. http://www.w3.org/TR/sawsdl/.
5. Miguel García, Jose María Álvarez, Diego Berrueta, and Luis Polo. Declarative Data Grounding Using a Mapping Language. *Communications of SIWN*, 2009.
6. Nicolai M. Josuttis. *SOA in Practice: The Art of Distributed System Design (Theory in Practice).* O'Reilly Media, Inc., August 2007.

[8] http://prima.morfeo-project.org/

7. David Martin, Mark Burstein, et al. OWL-S: Semantic Markup for Web Services. W3C member submission, W3C, 2004. http://www.w3.org/Submission/OWL-S/.
8. E. Motta, J. Domingue, L. Cabral, and M. Gaspari. IRS-II A Framework and Infrastructure for Semantic Web Services. *ISWC 2003. LNCS.*, 2870:306–318, 2003.
9. Michal Zaremba, Matthew Moran, and Thomas Haselwanter. WSMX Architecture. Final draft, DERI, 2005. Available at: http://www.wsmo.org/TR/d13/d13.4/v0.2/.