Using the Web Service Modelling Ontology to enable Semantic eBusiness*

Jos de Bruijn[1] Dieter Fensel[1] Uwe Keller[1]

Rubén Lara[2,†]

[1] *Digital Enterprise Research Institute*

{jos.debruijn, dieter.fensel, uwe.keller}@deri.org

[2] *Tecnología, Información y Finanzas*

rlara@afi.es

December 6, 2005

Traditionally, integration of applications, as well as integration of businesses, is achieved using costly customized solutions for every pair of applications and every pair of businesses. This requires a business to invest in custom hardware and software infrastructure for each new business partner, not to mention the effort required on a human level to agree upon data formats and interaction protocols, as well as Service Level Agreements (SLAs). For these reasons, the degree of re-usability of current integration solutions is remarkably low [3].

In this context, the Web provides an existing and highly available infrastructure for connecting business partners anywhere and anytime. In addition, Web services [2] provide a set of standards for the provision of functionality over the Web; the specification of the Simple Object Access Protocol (SOAP) as a standard for transmitting messages and of the Web Services Description Language (WSDL) as a standard for describing interfaces provide platform-independent access to back-end functionality. Together, the Web infrastructure and Web Service descriptions have the potential of reducing the cost of integrating applications and integrating business partners, because no custom communication lines need to be built, no proprietary messaging protocols need to be implemented, and no proprietary descriptions of how to communicate with applications and business partners need to be implemented and interpreted.

However, Web Service descriptions in WSDL are purely syntactic. The consumer of a service has to rely on a human-language description of the Web service to decide whether the Web service offers the desired functionality. Furthermore, there is no way to cooperate if both business partners have different

†Author names are in alphabetical order

interaction styles or use different terms for the description of the (desired and offered) data formats and functionality. Therefore, the human programmer needs to be kept in the loop for the manual selection of Web services, and custom software needs to be implemented to interact with the selected Web services.

A higher degree of automation in the location and use of Web Services can be achieved by adding explicit semantics to Web Service descriptions. Such semantically enriched descriptions are usually referred to as Semantic Web Services [9], and they are expected to enable businesses to dynamically locate partners which provide particular services, and to facilitate (semi-)automated cooperation with them.

The Web Service Modeling Ontology (WSMO) [10] follows this direction and defines an explicit conceptual model for Semantic Web Services based on the Web Service Modelling Framework (WSMF) [5]. The major aim of WSMO is to provide the necessary technology to achieve flexible and cost-effective integration within and across business boundaries.

For this purpose, WSMO defines four basic conceptual modelling elements which are required to achieve flexible integration, namely:

Ontologies Ontologies [4] provide formal and explicit specifications of the vocabularies used by the other modelling elements. Such formal specifications enable automated processing of WSMO descriptions and provide background knowledge for goal and Web Service descriptions.

Goals Goals describe the objectives a business might have when searching a business partner for cooperation.

Web Service descriptions Web Service descriptions formally describe services provided by businesses, i.e. they describe the provision of value such businesses offer. They also specify means of interacting with the provider in order to achieve the requested service. In addition, they contain a description of how the provider (dynamically) cooperates with other business partners to provide the offered service to the requester.

Mediators The dynamic cooperation between businesses can bring several sources of heterogeneity: goals and Web Services might use different ontologies i.e. vocabularies, and different business partners might use different interaction styles, or business protocols, as well as differences in the business process guiding such interactions. Mediators (cf. connectors in software architecture [1]) resolve such differences and enable seamless integration of business partners, overcoming heterogeneity in vocabularies, protocols and processes.

Challenges of integration

There are three major types of differences in applications and businesses which hamper integration [5]:

Differences in Vocabularies Different businesses describe desired and provided functionality using a different terminology. Such differences are typically not made explicit and are thus hard to deal with. Through ontologies, WSMO provides means to formally and explicitly specify the vocabulary used by an organization (or even a single application) [4]. This formal and explicit specification enables detecting differences and overlap between vocabularies used by different business partners and resolving conflicts between such vocabularies.

Differences in Protocols Different applications and different business partners have different styles of interaction. For example, one partner might implement a particular business protocol prescribed by RosettaNet, whereas another partner uses Electronic Business using eXtensible Markup Language (ebXML).

Differences in Business Processes Different businesses have different business processes running inside their organization. Such differences hamper inter-operation.

Summarizing, the major obstacles in automated business integration are heterogeneity in vocabularies, protocols and processes. WSMO introduces the concepts of mediators to overcome these differences.

Mediation in WSMO

As business cooperations can be dynamically established without prior agreement, the differences in data, process and protocol between the business involved in the cooperation can be expected.

For a service description framework to be usable in a real setting, the resolution of the heterogeneity inherent to a distributed and dynamic environment has to be considered explicitly. Mediators serve this purpose by connecting heterogeneous business partners and overcoming such heterogeneities in order to enable smooth integration.

WSMO defines different types of mediators, corresponding to the links which can be established between the different WSMO elements:

- *ooMediators* link ontologies to ontologies and other WSMO elements and resolve differences and conflicts between ontologies.
- *wwMediators* link services provided by a given business to other services they depend on. They resolve process and protocol differences. *wwMediators* resolve differences in vocabularies through the use of *ooMediators*.
- *wgMediators* link goals and Web Services and resolve protocol and process differences between the requester and the provider. *wgMediators* use *ooMediators* to overcome differences in the vocabularies used by the connected goals and services.

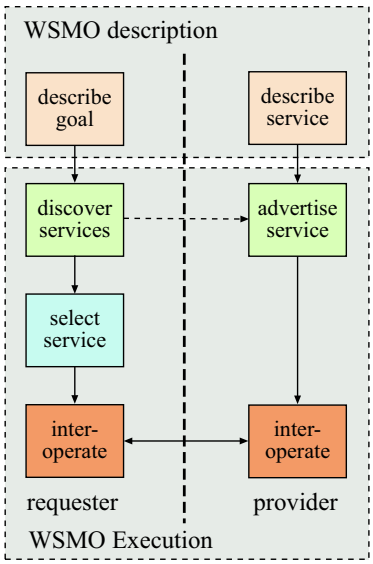

Figure 1: The usage of WSMO by the requesting and the providing business partners

- *ggMediators* connect goals and allow refinement of goals (cf. [6]), thereby allowing to specify generic, reusable, goals.

The description of reusable mediators enables establishing dynamic business relations, as no customized infrastructure is required for each business partner. Instead, the mediator infrastructure of WSMO can be used for efficient and dynamic integration.

Mediators are expected to link reusable vocabularies and interaction styles, thus providing a reusable solution to the integration of heterogeneous businesses. Therefore, WSMO enables seamless inter-operation with many different businesses, eliminating the costly dependency of a given business on few business partners. Rather, because the cost of locating new business partners and inter-operating with them is drastically reduced[1], it is possible to integrate with many different businesses, optimizing the cost efficiency (or other properties such as reliability or security) of the required services. Furthermore, the robustness of business processes is increased, as any service used can be automatically replaced in case of failure.

Usage of WSMO

Figure 1 illustrates how different business partners can exploit the WSMO framework to interact in a flexible, dynamic way. The left side of the figure depicts the process of finding a business partner offering a desired service. The right side of the figure depicts the process of describing and offering a particular service. Most notably, we do not assume that potential partners are already identified and predefined outside the WSMO framework; instead, in WSMO, goals and services are loosely coupled in order to allow all parties to interact in a flexible and cost-effective manner.

Requester-side Process

Describing the goal The requester formally describes the desired functionality of the service using terms from an ontology. *ggMediators* can be used to specify refinement of goals, enabling a hierarchical organization of goals (cf. refinement of tasks in [6]).

Discover services The goal that has been described by the requester is matched with different service descriptions created by different businesses which offer services. Mediators help in this discovery process by relating different vocabularies used to describe the goal and the different services.

Select service Once a number of services offering the desired functionality have been discovered, the actual service to interact with has to be selected, service level agreements have to be established and, if no existing mediators can be reused, mediators need to be constructed. This step often required interaction with the service provider. Outcome of this step is the selection of the actual service(s) to inter-operate with, service level agreements regarding the service being offered and a set of mediators required for seamless inter-operation.

Inter-operation After selection of the service, the service is executed and the business partners cooperate. Mediators help by translating data between different representations and by overcoming differences in interaction styles.

Provider-side process

Describe Web Service A service is described in terms of its functionality (the *capability*) and in terms of its interface (the *choreography* and *orchestration*).

Capability The capability of a service is a formal description of the functionality of a service in terms of:

[1]The cost is low because the same infrastructure can be used for interaction with different partners.

- *assumptions*, which are requirements on the state of the world before execution of the service.
- *preconditions*, which are requirements on the state of the information space before execution of the service.
- *effects*, which describe the state of the world after execution of the service.
- *postconditions*, which describe the state of the information space after execution of the service.

Choreography The choreography describes the interaction style of the service. It formally specifies which kind of messages are expected as input to the service and which kind of messages can be expected as output from the service. The choreography is the interface of the service exposed to requesters of the service.

Orchestration The orchestration describes the way the service uses other services to achieve its functionality. The orchestration can be seen as an interface of the service to other service providers. When reusing other services, the provider of the service acts as requester of other services in the WSMO usage process.

Conceptually, the choreography and orchestration are both decompositions of the functionality described in the capability; they describe ways of achieving the functionality of the service. Figure 2 illustrates the relation between the capability, choreography and orchestration. The figure also illustrates how a service reuses another (external) service through its orchestration.

Advertise service In order to enable businesses to discover the service, the service needs to be advertised, typically through a service repository.

Inter-operate The inter-operation step in the provider process is the same as the inter-operation step in the requester process.

Related Work

OWL-S [8] is an ontology for semantically describing services. The major difference to WSMO is that OWL-S does not consider the resolution of heterogeneity explicitly. While WSMO includes mediators as one of its key conceptual elements, OWL-S assumes that heterogeneity will be overcome by the underlying service infrastructure. For a complete comparison of both initiatives see [7].

METEOR-S[2] aims at integrating current (syntactical) Web Service initiatives for description, composition, etc. with Semantic Web technologies. However, METEOR-S does not provide a conceptual model for the description of business services and does not specifically address the integration of heterogeneous businesses.

[2]http://lsdis.cs.uga.edu/Projects/METEOR-S/

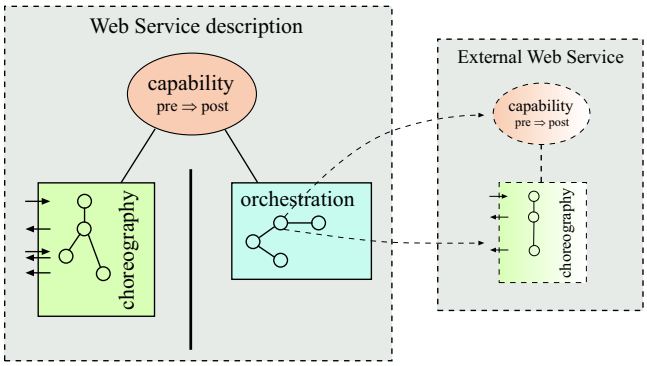

Figure 2: WSMO Service Description

Conclusions

Current eBusiness solutions often require a costly and custom hardware and software infrastructure for each pair of cooperating business partners. Furthermore, the lack of formal descriptions of services offered by organizations hampers automation in the location and usage of services required to perform a given business activity. Web Services provide a uniform infrastructure for the provision of services leveraging Web technologies, but they offer only syntactical descriptions that are hardly amenable to automation. Semantic Web Services enable the formal specification of services, allowing their automated, goal-driven, location and usage. WSMO provides a framework for the description of Semantic Web Services which enables seamless business integration through formal descriptions, maximal decoupling of components, and strong mediation support.

References

[1] Robert Allen and David Garlan. A Formal Basis for Architectural Connection. *ACM Transactions on Software Engineering and Methodology*, 6(3):213–249, July 1997.

[2] Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. *Web Services*. Springer-Verlag, Berlin Heidelberg, 2004.

[3] Christoph Bussler. *B2B Integration*. Springer-Verlag, Berlin, 2003.

[4] Dieter Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce, 2nd edition*. Springer-Verlag, Berlin, 2003.

[5] Dieter Fensel and Christoph Bussler. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, 1(2):113–137, 2002.

[6] Dieter Fensel and Enrico Motta. Structured Development of Problem Solving Methods. *Transactions on Knowledge and Data Engineering*, 13(6):913–932, 2001.

[7] Rubén Lara, Axel Polleres, Holger Lausen, Dumitru Roman, Jos de Bruijn, and Dieter Fensel. A Conceptual Comparison between WSMO and OWL-S. Final Draft D4.1v0.1, WSMO, 2005. http://www.wsmo.org/TR/d4/d4.1/v0.1/.

[8] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, Katia Sycara, *OWL Web Ontology Language for Services (OWL-S)*. W3C Member Submission, 22 November 2004. http://www.w3.org/Submission/OWL-S/.

[9] Sheila McIlraith, Tran Cao Son, and Honglei Zeng. Semantic Web Services. *IEEE Intelligent Systems, Special Issue on the Semantic Web*, 16(2):46–53, 2001.

[10] Dumitru Roman, Uwe Keller, Holger Lausen, Jos de Bruijn, Ruben Lara, Michael Stollberg, Axel Polleres, Dieter Fensel, and Christoph Bussler. Web Service Modeling Ontology. *Applied Ontology Journal*, 2005.