

By Jos de Bruijn, Dieter Fensel, Uwe Keller, and Rubén Lara

USING THE WEB SERVICE MODELING ONTOLOGY TO ENABLE SEMANTIC E-BUSINESS

The Web infrastructure offers businesses a way to partner and work together, integrating differing applications and communications in a seamless manner.

The integration of business applications is traditionally achieved using costly customized solutions for every set of applications for every business. Indeed, it requires a business to invest in a custom hardware and software infrastructure for each new 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 reusability 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 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 customized communication lines must be built, no proprietary messaging protocols must be implemented, and no proprietary descriptions of how to communicate with applications and business partners must be implemented and interpreted.

However, Web Service descriptions in WSDL are purely syntactic. The consumer of a service must rely on a human-language description of the Web service to decide whether it offers the desired functionality. Furthermore, there is no way to cooperate if both business partners have different interaction styles or use different terms for the description of the (desired and offered) data formats and functionality. Therefore, the programmer must be kept in the loop for the manual selection of Web services, and custom software must 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], which are expected to enable businesses to dynamically locate partners that 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 Modeling Framework (WSMF) [5]. The 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 modeling elements required to achieve flexible integration, namely:

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

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

Web Service descriptions formally describe services provided by businesses, that is, they describe the provision of value such businesses offer. They also specify the 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, or 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 (connectors in software architecture [1]) resolve such differences and enable seamless integration of business partners, overcoming heterogeneity in vocabularies, protocols, and processes.

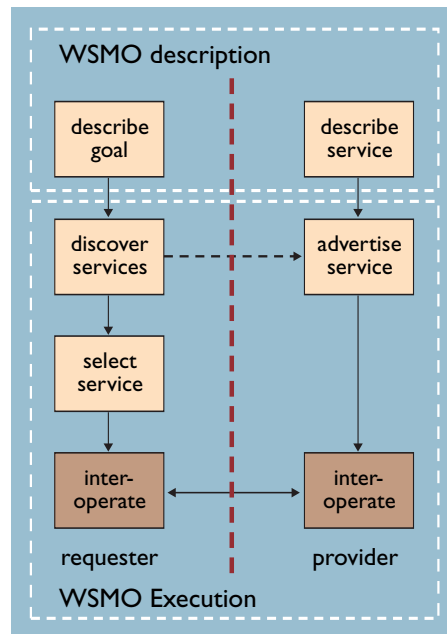


Figure 1. The usage of WSMO by the requesting and the providing business partners.

CHALLENGES OF INTEGRATION

There are three major types of differences in applications and businesses that hamper integration

[5]:

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

Differences in protocols. Business partners have their own style of interaction. For example, one partner might implement a particular business protocol prescribed by RosettaNet, whereas another employs Electronic Business using eXtensible Markup Language (eXML).

Differences in business processes. Businesses have different business processes running inside their organization. Such differences hamper interoperation.

Given these obstacles in automated business inte-

gration, WSMO introduces the concepts of mediators to overcome these differences.

MEDIATION IN WSMO

As business cooperation can be dynamically established without prior agreement, the differences in data, process, and protocol between cooperating businesses 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 must 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 that 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.

ggMediators connect goals and allow refinement of goals [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 interoperability with many different businesses, eliminating the costly dependency of a given business on few business partners. Because the cost of locating new business

partners and interoperating with them is drastically reduced,¹ it is possible to integrate with many 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

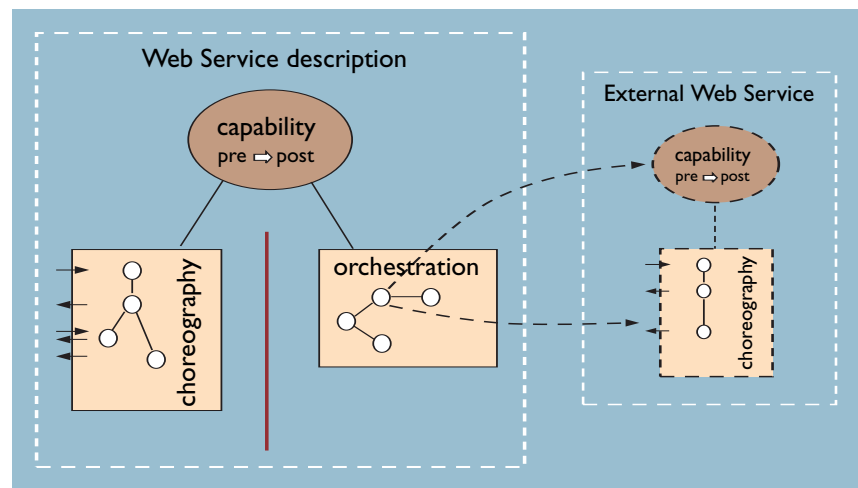


Figure 2. WSMO service description.

flexible, dynamic way. The left side of the figure depicts the process of finding a business partner offering a desired service. The right side illustrates 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 goals and services in WSMO are loosely coupled in order to allow all parties to interact in a flexible, 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 (refinement of tasks in [6]).

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

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

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 that enables seamless business integration through formal descriptions, maximal decoupling of components, and strong mediation support.

Select service. Once a number of services offering the desired functionality have been discovered, the actual service to interact with must be selected, SLAs must be established, and if no existing mediators can be reused, mediators must be constructed. This step often required interaction with the service provider. The outcome is the selection of the actual service(s) to interoperate with, SLAs regarding the service being offered, and a set of mediators required for seamless interoperation.

Interoperation. After the service is selected, it is executed and the business partners can 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 interface (the choreography and orchestration).

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

- 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.
- Post-conditions, which describe the state of the information space after execution of the service.

Choreography describes the interaction style of the service. It formally specifies what kind of messages are expected as input to the service and what kind of mes-

sages can be expected as output. The choreography is the interface of the service exposed to requesters of the service.

Orchestration describes the way the service uses other services to achieve its functionality; it can be seen as an interface to other service providers. When reusing other services, the provider acts as the 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. It 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 must be advertised, typically through a service repository.

Interoperate. The interoperation step in the provider process is the same as the interoperation step in the requester process.

RELATED WORK


OWL-S [8] is an ontology for semantically describing services. Its 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 (lstdis.cs.uga.edu/Projects/METEOR-S/) aims to integrate current (syntactical) Web Service initiatives for description, composition, and so on,

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.

CONCLUSION

Current e-business 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 that enables seamless business integration through formal descriptions, maximal decoupling of components, and strong mediation support. 

REFERENCES

1. Allen, R. and Garlan, D. A formal basis for architectural connection. *ACM Trans. Softw. Eng. Methodology* 6, 3 (July 1997), 213–249.
2. Alonso, G., Casati, F., Kuno, H., and Machiraju, V. *Web Services*. Springer-Verlag, Berlin Heidelberg, 2004.
3. Bussler, C. *B2B Integration*. Springer-Verlag, Berlin, 2003.
4. Fensel, D. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*, 2nd ed. Springer-Verlag, Berlin, 2003.
5. Fensel, D. and Bussler, C. The web service modeling framework (WSMF). *Electr. Commerce Res. Apps* 1, 2 (2002), 113–137.
6. Fensel, D. and Motta, E. Structured development of problem-solving methods. *Trans. Knowledge and Data Engineering* 13, 6 (2001), 913–932.
7. Lara, R., Polleres, A., Lausen, H., Roman, D., de Bruijn, J., and Fensel, D. A conceptual comparison between WSMO and OWL-S (2005); www.wsmo.org/TR/d4/d4.1/v0.1/.
8. Martin, D. et al. OWL Web ontology language for services (OWL-S). W3C Member Submission (Nov. 22, 2004); www.w3.org/Submission/OWL-S/.
9. McIlraith, S., Cao Son, T., and Zeng, H. Semantic Web services. *IEEE Intelligent Systems* 16, 2 (2001), 46–53.
10. Roman, D. et al. Web service modeling ontology. *Applied Ontology J.* (2005).

JOS DE BRUIJN (jos.debruijn@deri.org) is a researcher at the Digital Enterprise Research Institute, Innsbruck, Austria.

DIETER FENSEL (dieter.fensel@deri.org) is a professor at the Digital Enterprise Research Institute, Innsbruck, Austria, and Galway, Ireland.

UWE KELLER (uwe.keller@deri.org) is a researcher at the Digital Enterprise Research Institute, Innsbruck, Austria.

RUBÉN LARA (rlara@afi.es) is R&D director at Tecnología, Información y Finanzas, Madrid, Spain.

© 2005 ACM 0001-0782/05/1200 \$5.00