

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <<http://www.upgrade-cepis.org/>>

Publisher

UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <<http://www.cepis.org/>>) by **Novática** <<http://www.ati.es/novatica/>>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*, <<http://www.ati.es/>>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by **Novática**

UPGRADE was created in October 2000 by CEPIS and was first published by **Novática** and **INFORMATIK/INFORMATIQUE**, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <<http://www.svifsi.ch/>>)

UPGRADE is the anchor point for **UPENET** (UPGRADE European NETWORK), the network of CEPIS member societies' publications, that currently includes the following ones:

- **Informatik-Spektrum**, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- **ITNOW**, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- **Mondo Digitale**, digital journal from the Italian CEPIS society AICA
- **Novática**, journal from the Spanish CEPIS society ATI
- **OCG Journal**, journal from the Austrian CEPIS society OCG
- **Pliroforiki**, journal from the Cyprus CEPIS society CCS
- **Pro Dialog**, journal from the Polish CEPIS society PTI-PIPS

Editorial Team

Chief Editor: Llorenç Pagés-Casas, Spain, <pages@ati.es>

Associate Editors:

François Louis Nicolet, Switzerland, <nicolet@acm.org>

Roberto Carniel, Italy, <carniel@dgt.uniud.it>

Zakaria Maamar, Arab Emirates, <Zakaria.Maamar@zu.ac.ae>

Soraya Kouadri Mostéfaoui, Switzerland, <soraya.kouadrimostefaoui@gmail.com>

Rafael Fernández Calvo, Spain, <rfcvalvo@ati.es>

Editorial Board

Prof. Wolfried Stucky, CEPIS Former President

Prof. Nello Scarabottolo, CEPIS Vice President

Fernando Piera Gómez and

Llorenç Pagés-Casas, ATI (Spain)

François Louis Nicolet, SI (Switzerland)

Roberto Carniel, ALSI - Tecnoteca (Italy)

UPENET Advisory Board

Hermann Engesser (Informatik-Spektrum, Germany and Switzerland)

Brian Runciman (ITNOW, United Kingdom)

Franco Filippazzi (Mondo Digitale, Italy)

Llorenç Pagés-Casas (Novática, Spain)

Veith Risak (OCG Journal, Austria)

Panicos Masouras (Pliroforiki, Cyprus)

Andrzej Marciniak (Pro Dialog, Poland)

Rafael Fernández Calvo (Coordination)

English Language Editors: Mike Andersson, Richard Butchart, David Cash, Arthur Cook, Tracey Darch, Laura Davies, Nick Dunn, Rodney Fennemore, Hilary Green, Roger Harris, Michael Hird, Jim Holder, Alasdair MacLeod, Pat Moody, Adam David Moss, Phil Parkin, Brian Robson

Cover page designed by RFCalvo, © Rafael Fernández Calvo 2006

Layout Design: François Louis Nicolet

Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>

Advertising correspondence: <novatica@ati.es>

UPGRADE Newslist available at

<<http://www.upgrade-cepis.org/pages/editinfo.html#newslist>>

Copyright

© Novática 2006 (for the monograph)

© CEPIS 2006 (for the sections MOSAIC and UPENET)

All rights reserved under otherwise stated. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, contact the Editorial Team

The opinions expressed by the authors are their exclusive responsibility

ISSN 1684-5285

Monograph of next issue (December 2006) "Open Document Format (ODF)"

(The full schedule of UPGRADE
is available at our website)



The European Journal for the Informatics Professional
<http://www.upgrade-cepis.org>

Vol. VII, issue No. 5, October 2006

Monograph: Web Services (published jointly with Novática*)

Guest Editors: *José Carlos del Arco-Prieto, Jesús Arias-Fisteus, Óscar Corcho-García, Jorge Cardoso*

- 2 Presentation. Web Services from An Industrial and An Academic Perspective — *Jose Carlos del Arco-Prieto, Jesús Arias-Fisteus, Óscar Corcho-García and Jorge Cardoso*
- 5 Web Services: Introduction and State of The Art — *Óscar Corcho-García, José Carlos del Arco-Prieto and Jesús Arias-Fisteus*
- 11 W3C and Web Services Standardization — *Martín Álvarez-Espinar*
- 16 Integration and Interoperability Experiences in Healthcare — *Antonio García-Landeira*
- 21 Model-Driven Extra-Functional Property Development for Web Services: A Case Study from The Service and Client Side Perspectives — *Guadalupe Ortiz-Bellot and Juan Hernández-Núñez*
- 29 Towards an Automated Trading Process — *Pablo Fernandez-Montes, Manuel Resinas-Arias de Reyna and Rafael Corchuelo-Gil*
- 34 Semantic Web Services with WSMO — *Holger Lausen, Jos de Bruijn, Uwe Keller and Rubén Lara*
- 38 Towards Semantic Service Selection for B2B Integration — *Andreas Friesen and Kioumars Namiri*
- 46 Leveraging E-Marketplace Models for Web Service-Based Application Development — *Abraham Nieva de la Hidalga, Liping Zhao and Pedro R. Falcone-Sampaio*

UPENET (UPGRADE European NETWORK)

- 52 From **Novática** (ATI, Spain)
Open Standards
Open Standards, A Renewed Impulse: The Case of Open Document Format — *Miguel A. Amutio-Gómez*
- 59 From **Mondo Digitale** (AICA, Italy)
ICT Security
Critical Information Infrastructure Protection — *Sandro Bologna, Roberto Setola and Salvatore Tucci*
- 68 From **ITNow** (BCS, United Kingdom)
Informatics Profession
Professionalizing IT — *Peter Skyte*

CEPIS NEWS

- 70 Harmonise Project: Presentation

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <<http://www.ati.es/novatica/>>.

Semantic Web Services with WSMO

Holger Lausen, Jos de Bruijn, Uwe Keller and Rubén Lara

The integration of applications is one of the main challenges when building IT solutions. Integration is often achieved using costly customized solutions for every pair of applications. Web service technologies are a set of standards that allow software interfaces to be defined using XML (eXtensible Markup Language) as the message format and the Internet infrastructure for message transport. While lowering costs, Web service technologies by themselves do not ensure that two businesses use the same data structures or business protocols, neither do they provide the means to resolve potential conflicts. 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. Semantic Web Services enable the formal specification of services, allowing their automated, goal-driven location and usage. The Web Service Modelling Ontology (WSMO) provides a framework for the description of Semantic Web services which enables seamless integration through formal descriptions, maximal decoupling of components, and strong mediation support.

Keywords: Semantic Web Services, Service Composition, Service Discovery, Service Invocation, Service Negotiation, Web Service Modelling Ontology.

1 Introduction

One of the main challenges for the IT industry today is the integration of applications. Traditionally, integration needs mainly occurred within one enterprise between dif-

ferent systems such as human resources and accounting. Nowadays integration crosses the boundaries of different enterprises. Typically integration is achieved using costly customized solutions for every pair of applications and every pair of businesses. This requires an enterprise to invest in custom infrastructure for each new business partner, not to mention the effort required on a human level to agree upon data formats and interaction protocols. For these reasons,

Authors

Holger Lausen obtained his diploma in Computer Science at Flensburg University of Applied Science in 2003. His diploma thesis discussed the integration of semantic web technologies and document management systems. Before he joined the *Digital Enterprise Research Institute* (DERI) as a researcher in April 2003, he was involved in various software development projects in Germany and abroad. In DERI, he was project coordinator for the *Semantic Web Enabled Web Services* (SWWS) project, a pioneer European research project in the field of Semantic Web Services, and he now works in the Knowledge Web network of excellence and the FIT-IT project Reasoning with Semantic Web Services. He is active in the field of ontology and language design for the annotation of Web Services. He is the editor of the W3C specification Semantic Annotation for WSDL and a member of several Web Service related working groups. <holger.lausen@deri.org>.

Jos de Bruijn received his Master degree in Technical Informatics from the Delft, University of Technology, the Netherlands, in 2003. Since 2003 he has been employed as a researcher at the *Digital Enterprise Research Institute* (DERI), at the University of Innsbruck, Austria. His research interests include Semantic Web (Services) (languages), Logical languages, Logic Programming, and Non-Monotonic Reasoning. He is the author of 15+ peer-reviewed publications in the area of Semantic Web (Services) languages, including several journal publications. He has been actively involved in the European funded projects COG, SEKT, DIP, and Knowledge Web, and has been the project lead for SEKT at DERI. He is a member of

the WSMO and WSML working groups and of the W3C RIF working group. <jos.debruijn@deri.org>.

Uwe Keller is a researcher in semantic technologies and their applications at the *Digital Enterprise Research Institute* (DERI), Leopold Franzens University, Innsbruck, Austria. His current research interests include logics and automated reasoning with a specific focus on Semantic Web applications, semantic description of Web services and the exploitation of such descriptions in applications, knowledge-based applications, formal methods and semantically-enriched service-oriented architectures. Uwe obtained a masters degree in Computer Science at the University of Karlsruhe (TH), Germany. <uwe.keller@deri.org>.

Rubén Lara is R&D director at Grupo Analistas, where he leads several projects on the exploitation of semantics in Service-Oriented Architectures, and on the application of semantic technologies to the financial domain. Rubén Lara obtained his MSc in computer Science at the *Universidad Autónoma de Madrid*, and received the First National Award in Computer Science from the Spanish Ministry of Science and Education, as well as the Special Award in Computer Science from the *Universidad Autónoma de Madrid*. Before joining Grupo Analistas, he worked at the Digital Enterprise Research Institute in Austria and Ireland as a researcher on semantic Web and semantic Web services, and was the managing director of the EU-funded network of excellence Knowledge Web. Rubén has authored several publications on international journals and conferences, and has co-authored the book "*Information integration with ontologies: experiences from an industrial showcase*". <rlara@afi.es>.

the degree of re-usability of current integration solutions is remarkably low [3].

Web service technologies [2] are a set of standards that allow software interfaces to be defined using XML (*eXtensible Markup Language*) as the message format and the internet infrastructure for message transport. Many vendors have adopted Web services, so a variety of tools for various platforms is available. Web Services have the potential of reducing the cost of integrating applications 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 services technologies by themselves do not ensure that two businesses use the same data structures or business protocols and they do not provide the means to resolve such conflicts.

While Web service technologies focus on the technical description of interface, the *Service-oriented Architecture* (SOA) paradigm provides guidelines on how to design systems, so that the integration of new components and the rearrangement of existing components can be simplified. Essentially, the SOA paradigm prescribes a loose coupling between components. The selection of a particular service should not be limited to the design time of a system, but the architecture enables services to be selected at runtime. Thereby services can be distributed with an explicit network component (e.g. Internet infrastructure). Web service technologies can be used to implement an SOA. However, Web service descriptions in WSDL are of limited expressivity. 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 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.

2 Limitations of Current Technologies

In a software-oriented architecture a number of central tasks have been identified. These include the discovery of new services, the negotiation of specific business terms with a given service, as well as the composition and invocation

```
instance EAN memberOf ProductCode
instance UPC memberOf ProductCode
concept barCodeScanner
canProcess ofType ProductCode
axiom definedBy
?scanner[canProcess hasValue EAN] implies
?scanner[canProcess hasValue UPC].
```

Figure 1: An Ontology Example to Illustrate how They Can Aid Integration.

of atomic services to achieve a more complex functionality. This list of tasks is not exhaustive; in fact there are many more tasks. However, we will use these to illustrate the limitations of current technologies and later to show how WSMO (*Web Service Modelling Ontology*) can be used to overcome some of these limitations.

Currently the discovery of new services can be done using keyword based retrieval or using predefined taxonomies. While the former lacks the necessary precision for automating the process, the latter is inflexible since the taxonomy is maintained centrally and each change or addition requires a lengthy standardization process. Alternatively customers can try to find providers based on a given interface definition, however, the fact is providers do not use standardized interfaces, but have variations in the interfaces even if they offer similar functionality. Generally speaking, we can observe three major types of differences which hamper integration [5]:

1. Differences in Vocabularies i.e. different data formats for a similar data entity such as an address.
2. Differences in Protocols which are different styles of interaction, e.g. a synchronous reply to a purchase order (like for a typical online shop) versus the multiple asynchronous replies that occur in a RosettaNet¹ Purchase Order.
3. Different businesses may have differences in business processes running inside their organization, e.g. one business sells articles at fixed prices, whereas the other operates its business according to an auction model where multiple orders are gathered and only the most profitable are chosen.

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 [7], and they are expected to enable businesses to dynamically locate partners which provide particular services, and to facilitate (semi-)automated cooperation with them. Semantic descriptions add machine-processable semantics to data. The computer can "understand" the information and therefore process it on behalf of the human user. The Web Service Modelling Ontology provides a conceptual model how to formalize the semantics of Web services.

3 Web Service Modelling Ontology

The *Web Service Modelling Ontology* (WSMO) [6] describes all relevant aspects related to services that are accessible through a Web service interface. Its ultimate goal is to enable the (total or partial) automation of the tasks (e.g. discovery, selection, composition, mediation, execution, and monitoring) involved in both intra- and inter-enterprise integration of Web services. WSMO has its conceptual basis in the *Web Service Modelling Framework* (WSMF) [5]; it refines its concepts and formalizes them as

¹ <http://www.rosettanet.org/>

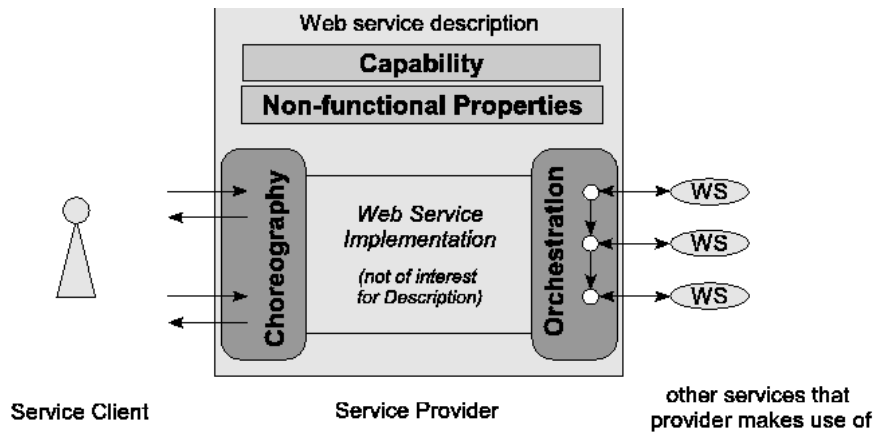


Figure 2: Structure of A WSMO Web Service Definition

ontology. For this purpose, WSMO defines four basic modelling elements, namely:

Ontologies [4] provide formal and explicit specifications of the vocabularies used by the other modelling elements. Ontologies form the backbone of the Semantic Web; they allow machine understanding of information through the links between the information resources and the terms in the ontologies. Furthermore, ontologies facilitate interoperation between information resources through links to the same ontology or links between ontologies. A key feature of ontologies is that, through formal, real-world semantics and consensual terminologies, they interweave human and machine understanding of symbols. These symbols, also called terms and relations, can be interpreted by both humans and machines. The meaning for a human is represented by the term itself, which is usually a word in natural language, and by the semantic relationships between terms. In Figure 1 we give a brief example that illustrates how ontologies can aid integration.

In that example we formalize aspects of commonly used product codes, i.e. the EAN (*European Product Number*) and the UPC (*Universal Product Code*). First we establish that both are product codes, secondly we declare the concept of a bar code scanner and one of its properties (i.e. that it can read a certain product code). In fact it turns out that every scanner that can process EAN product codes can also process UPC bar codes. This knowledge has been formalized in the axiom at the bottom of the example. Once declared, this knowledge can be used, e.g. to determine during a purchase order process that, although one expects an article with EAN code, one with UPC can be accepted as well.

Web service descriptions document and formalize services provided by businesses. As illustrated in Figure 2, different elements describe various aspects. The value that the service can provide is captured by its capability. Within the capability one can declare for example that a particular service sells computer parts of certain brands. Non-functional properties can be used to describe aspects such as average response time or the encryption protocol used for the transaction. WSMO provides an extensible list of such properties that are based on common meta data standards such as Dublin Core [8].

The choreography describes how to interact with a service provider to request the actual performance of a service, or to negotiate aspects of its provision. A choreography defines the sequence in which different activities have to be performed; for example that for a purchase order first an order has to be created, then a number of line items can be added, and only after closing the order a confirmation about the availability will be returned.

The implementation that provides the service is transparent to us and not of interest for the description. We are concerned mainly with its style of interaction and only under certain circumstances with the services that are used to provide the value described in the capability; i.e. the way a service makes use of other services can be specified in the orchestration.

Goals describe the objectives when searching a Web service. They describe aspects related to the users desires with respect to the requested functionality. Goals are structured similarly to Web services and like all other elements of WSMO they use terms from ontologies in order to define all relevant aspects. Note that WSMO completely decouples the objectives that a requester has; i.e. his/her goal, from the services that can actually fulfil such a goal.

Mediators have a central role in WSMO, they enable differences in vocabularies, protocols and processes which may arise during the dynamic cooperation between services to be bridged. Mediators (cf. connectors in software architecture [1]) resolve such differences and enable seamless integration, overcoming heterogeneity in vocabularies, protocols, and processes. For each type of mediation, WSMO introduces different mediators, e.g. OO-Mediators (OO stands for *ontology to ontology*) resolves mismatches between vocabularies, and WW-Mediators focus on the mediation between different protocols.

4 Usage of WSMO

Earlier we outlined the upcoming paradigm of service-oriented architecture. The loose coupling between components enables the dynamic configuration of more complex functionality built from basic services. However with current Web service technologies the tasks involved mainly

have to be performed manually. The **different components** of WSMO can help to increase the level of automation through their formal character and the use of ontologies that interweave human understanding and machine processing.

Discovery: Before a service can be used in a distributed application, it must first be located. A service capabilities expresses the value a service provides. Unlike central taxonomies, the vocabulary to describe capabilities is formalized via decentralized ontologies. Interconnections in the form of logical axioms enable a given terminology to be extended and refined.

Negotiation: Whenever a suitable provider has been determined to serve a certain goal, it is necessary to negotiate a specific service. This includes the establishment of trust policies, the determination of payment method, etc. For the purpose of automating this task, it is important that a WSMO description not only specifies the functionality of a service (capability), but also includes non-functional aspects such as the price of a service.

Composition: In cases where a particular goal cannot be achieved by means of a single Web service, semantic descriptions can help to determine a promising combination of multiple Web services to achieve the requested functionality. A clever selection of services and lightweight planning or scheduling can ease the pain of manual integration. Composition requires not only the semantic annotation of the overall capabilities of a service, but also a behavioural description of how to interact with the Web service (choreography) in order to achieve a certain functionality. A composition that has been generated can be described using WSMO orchestration.

Invocation: After (a combination of) services have been selected, the final step is to invoke them. To this end, possible input and output values need to be extracted from the semantic goal description and adapted to the message formats and communication protocols that are expressed in the WSMO choreography.

We have outlined how the different components in WSMO can be used generally within the integration process. Let us now take a closer look into the different levels of abstraction in the descriptions that are required in the different steps. We go on to provide a proper conceptual grounding by acknowledging the difference between service discovery (finding a suitable provider in general) and Web service discovery (determining a specific service instance and the precise parameter of an interaction). It is important to understand that these two processes happen on different levels; while the former can be performed on static descriptions, the latter requires interaction with the service provider and is therefore much more complex.

Usually, a service provider does not provide just one specific service but rather a set of coherent and logically related services. For instance, an airline does not only provide the possibility to book a particular flight on a particular date, but instead it will offer (and advertise) the general service of booking flights. In order to deliver a specific service instance, a service provider usually needs certain infor-

mation from the requester. An airline might require the name of the person booking, the flight number, booking class, and a valid credit card number as input information in order to create a booking. This input data will determine which specific service will be provided. While the general service (booking flights) can be statically advertised in a WSMO capability (used for discovery), the WSMO choreography describes how to interact with the service, e.g. for the purpose of negotiating a concrete flight data. Negotiation also typically requires mediation, since the data structures and protocols of the client have to be mapped to those of the provider in order to interact with a provider to access, explore and consume specific services.

5 Conclusions

Current technologies require a costly and custom hardware and software infrastructure for each pair of cooperating business partners. 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 service technologies and the service-oriented architecture paradigm 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 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, 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, Holger Lausen, Axel Polleres, Jos de Bruijn, Michael Stollberg, and Jon Domingue. *Enabling Semantic Web Service, the Web Service Modeling Ontology*. Springer Verlag, 2005.
- [7] 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.
- [8] S. Weibel, J. Kunze, C. Lagoze, and M. Wolf. Dublin Core metadata for resource discovery. RFC 2413, IETF, September 1998.