



## D3.2 WSMO Use Case Modeling and Testing

WSMO Working Draft 08 October 2004

**This version:**

<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041008/>

**Latest version:**

<http://www.wsmo.org/2004/d3/d3.2/v0.1/>

**Previous version:**

<http://www.wsmo.org/2004/d3/d3.2/20041004/>

**Editors:**

Michael Stollberg  
Holger Lausen  
Axel Polleres  
Rubén Lara

**Authors:**

Michael Stollberg  
Holger Lausen  
Axel Polleres  
Rubén Lara  
Uwe Keller  
Michal Zaremba  
Armin Haller  
Dieter Fensel  
Michael Kifer

**Reviewer:**

Christoph Bussler

This document is also available in non-normative [PDF](#) version.

Copyright © 2004 [DERI](#)®, All Rights Reserved. [DERI](#) liability, trademark, document use, and software licensing rules apply.

---

### Abstract

This deliverable exemplifies the usage of the Web Service Modeling Ontology WSMO for modeling Web Service driven applications. In this document, we outline general usage of Semantic Web Service in different application fields, identifying the usage scenarios and the arising technical requirements. In additional documents, specific use cases are defined that

showcase and test different aspects around WSMO.

## Related Documents

WSMO Standard: [D2 v1.0 Web Service Modeling Ontology \(WSMO\)](#)

WSMO Primer: [D3.1 v0.1 WSMO Primer](#)

WSMO Reasoning: [D5.1 v0.1 WSMO Discovery](#)

WSMO Use Case: [D3.3 v0.1 Virtual Travel Agency](#)

WSMO Use Case: [D3.4 v0.1 B2B - Business Document Exchange](#)

WSMO Use Case: [D3.5 v0.1 Semantic Web Fred - Agent Collaboration with Semantic Web Services](#)

---

## Table of Contents

### **1. Introduction**

[1.1 Semantic Web Services](#)

[1.2 The Web Service Modeling Ontology WSMO](#)

### **2. Semantic Web Service Application Scenarios**

[2.1 B2C - Virtual Travel Agency](#)

[2.1.1 Description](#)

[2.1.2 Scope](#)

[2.1.3 Actors, Roles and Goals](#)

[2.1.4 Usage Scenarios](#)

[2.1.5 System Architecture](#)

[2.2 B2B - Integration with Semantic Web Services](#)

[2.2.1 Description](#)

[2.2.2 Scope](#)

[2.2.3 Actors, Roles and Goals](#)

[2.2.4 Usage Scenarios](#)

[2.2.5 System Architecture](#)

### **3. WSMO Use Cases**

[3.1. B2C - a Virtual Travel Agency for Online Train Tickets](#)

[3.2. B2B - Business Document Exchange](#)

[3.3. Semantic Web Fred - Agent Collaboration with Semantic Web Services](#)

### **4. Conclusions and Future Work**

[References](#)

[Acknowledgements](#)

[Appendix: Change Tracking](#)

---

## 1. Introduction

This document exemplifies the usage Semantic Web Services with special attention in usage and testing of the Web Service Modeling Ontology WSMO. We briefly replicate the objectives and the approach of WSMO and outline how specific use cases can be modeled in WSMO along with explanations on the modeling decisions. Then, we discuss several usage scenarios of Semantic Web Services, identifying the usage scenarios as well as the aims and challenges arising for Semantic Web Service technologies.

This Deliverable is intended to gather use cases for WSMO, and to evolve in accordance to the ongoing development of WSMO itself. The different use cases provided in subsequent documents serve as input and providing valuable insight for testing and adapting the modeling constructs provided in WSMO in real world scenarios for Web Services. So, besides demonstrating how to model Web Services in WSMO, the use cases also allow us to demonstrate the adequacy of our approach in terms of providing an exhaustive framework for covering all relevant aspects of semantic description of Web Services. In the long run, additional use cases will be added in order to widen possible solutions for Semantic Web Service technologies around WSMO.

This document is organized as follows: the remainder of [Section 1](#) summarizes the objectives and approach of WSMO; [Section 2](#) discusses possible application areas of Semantic Web Services. [Section 3](#) gathers different defined for and around WSMO. [Section 4](#) concludes the document. A Change Tracker in the [Appendix](#) explicitly list the major changes between different versions of this document in order to facilitate readers following the improvements.

## 1.1. Semantic Web Services

A Web Service is a piece of software accessible via the Internet. Current Web Service technologies allow exchange of messages between Web Services [[SOAP](#)], describing the technical interface [[WSDL](#)], and advertising a Web Services in a registry [[UDDI](#)]. These technologies do not provide any information about the meaning of information used, neither do they explicitly describe the functionality of a services as needed for automated usage and interoperability of Web Services. Enhanced Web Service technologies aim at more sophisticated techniques to describe Web Services, emphasizing the concept of Semantic Web Services. In our understanding, a Semantic Web Service is defined as a “self-contained, self-describing, semantically marked-up software resource that can be published, discovered, composed and executed across the Web in a task driven automatic way” [[Arroyo et al., 2004](#)]. In the end, by machine-processable descriptions of the relevant information of Web Services, the following tasks shall be addressed:

- **Automatic Web Service Discovery:** finding Web Services that can solve a goal defined by a service requester, i.e. a desire that a user wants to get resolved automatically by using Web Services.
- **Automatic Web Service Composition:** assembly of services based on their functional specifications in order to provide a functionality on a higher level.
- **Automatic Web Service Execution:** invocation of a concrete set of services, arranged in a particular way following programmatic conventions that realizes a given task.

## 1.2 The Web Service Modeling Ontology WSMO

The aim of the WSMO project is to define a coherent technology for Semantic Web Services (short: SWS). WSMO defines the modeling elements for describing several aspects of Semantic Web Services. The conceptual basis of WSMO is the Web Service Modeling

Framework [\[WSMF\]](#), wherein four main components needed for a full coverage framework for Semantic Web Services are defined (see Figure 1):

**Ontologies** provide the formal semantics of the information used by all other components.

**Goals** specify objectives that a client may have when it consults a web service.

**Web Services** exposed descriptions of the functionality of a Web Service for supporting automated discovery, composition, and execution (called “Capabilities” in WSMO). For supporting automated usage, composition, and execution of Web Services, particular information on the external visible behavior of a Web Service are specified (called “Interface” in WSMO), including information on the technical accessibility and the actual message exchange of Services.

**Mediators** are used as connectors between particular components and include possibly required mediation facilities needed to make connected components interoperable. WSMO distinguishes different types of Mediators.

Further details on the components of WSMO along with exhaustive explanations are presented in the WSMO Primer [\[Arroyo and Stollberg, 2004\]](#).

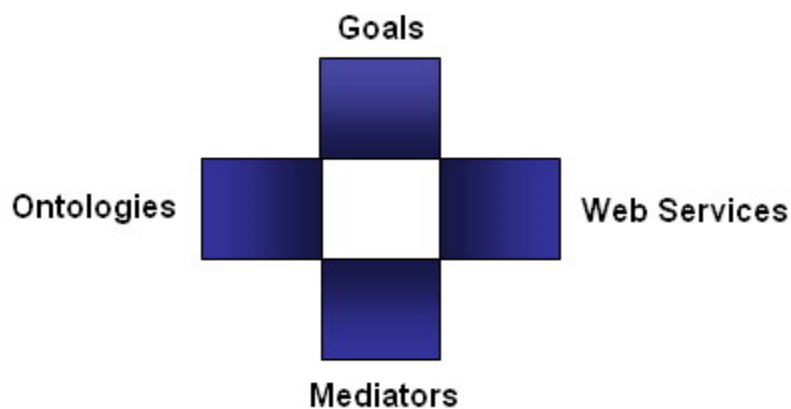


Figure 1. WSMO Components

## 2. Semantic Web Service Application Scenarios

Semantic Web Services can be used in manifold application fields. In accordance to the use cases defined in Web Services Architecture Usage Scenarios by the [W3C Web Services Architecture Working Group](#) (see [\[He et al., 2004\]](#)), we discuss two of the most commonly used scenarios to exemplify the usage of SWS technologies in this document:

1. In a “B2C” use case, i.e. a third party provides a service to end users acting as a Client aggregating other Semantic Web Services. Frequently mentioned examples of using Semantic Web Services within a B2C-setting refers to the travelling domain, wherein a “Virtual Traveling Agency” provides end-user services for e-Tourism by aggregating Web Services of different tourism service providers
2. The second example is concerned with B2B Integration wherein a business entity, e.g. a business document, is exchanged between enterprises. Therein, different aspects of EAI

might arise which shall be handled by Semantic Web Services technology.

For describing the use cases, we slightly modify the methodology of the W3C Use Case descriptions and extend them by the requirements arising for Semantic Web Services technologies. The aspects considered for our use case definitions are as follows:

- **Description:** describes the overall scenario
- **Scope:** defines the scope of the application scenario described
- **Actors, Roles and Goals:** identifies the actors in the scenario, their roles (i.e. what they do in the scenario) and their goals (i.e. what they want to achieve by participating in the scenario).
- **Usage Scenarios:** the W3C Service Architecture Working Group defines a [use case](#) as "... a sequence of interactions between a service requester and one or more services, which achieve measurable results for the requester", and a [usage scenario](#) as "... an atomic step in a path through a use case", i.e. an activity that has to be performed during execution of the use case and which can be automated by appropriate Semantic Web Service technologies. For each use case we describe the particular usage scenarios by the following informations:
  - participating actors and their goals
  - activities to be performed
  - technological requirements
  - possible extensions of the scenario.
- **System Architecture:** In addition to the use-case oriented aspects of the W3C methodology, we also outline the general requirements and possible architecture of the respective SWS-based application.

## 2.1 B2C - Virtual Travel Agency

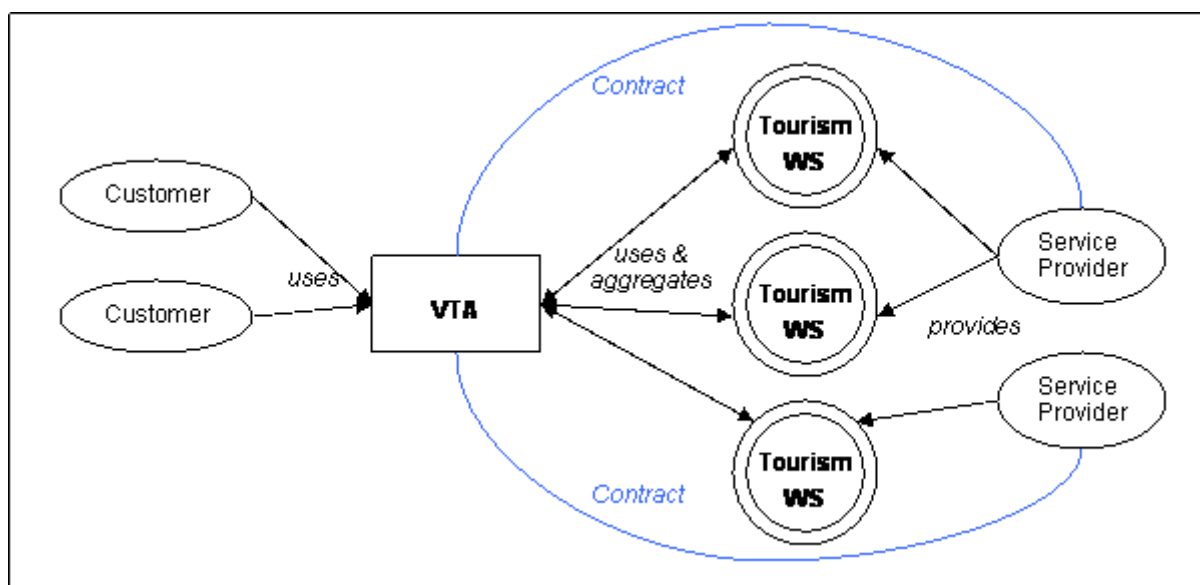
In [\[He et al., 2004\]](#), the travel agency use case is separated into two use cases - one with static discovery and one with automated discovery. With Semantic Web Services we clearly want to support automated discovery. Thus, in the first WSMO use case we will describe a Virtual Travel Agency example that involves automated discovery of Web Services.

### 2.1.1 Description

Imagine a "Virtual Traveling Agency", called VTA for short, which is an end user platform providing eTourism services to customers. These services can cover all kinds of information services concerned with tourism information - from information about events and sights in an area to services that support booking of flights, hotels, rental cars, etc. online. Such VTAs are already existent, but at this point mostly comprise simple information portals along with some web-based customer services. By applying Semantic Web Services, a VTA will invoke Web Services provided by several eTourism suppliers and aggregate them into new customer services in a (semi-)automatic fashion. Such VTAs providing automated eTourism services to end users thus tremendously enhance the functionality of currently existing VTAs.

Our VTA use case that aggregates Web Services of different tourism service providers in a nutshell shall provide the following functionality: A customer uses the VTA service as the entry point for his requests. These requests must fit to end-user services that the VTA provides. These end-user services are aggregated by the VTA by invoking and combining Web Services offered by several tourism service providers. Therefore, there must be some kind of contract

between the service providers and the VTA for regulating usage and allowance of the Web Services. Figure 2 gives an overview (modified and extended from [W3C Travel Agent Use Case overview](#), as defined in [\[He et al., 2004\]](#)).



**Figure 2. Use Case Overview: Virtual Travel Agency based on Semantic Web Services**

### 2.1.2 Scope

The scenario outlines a general structure for VTAs that can be extended to more complex scenarios wherein the customer can be a Web Service itself, thus creating a network of composed services that offer complex tourism services. For example, one VTA can provide flight booking services for an airline union, another VTA aggregates booking service for a worldwide hotel chain, and a third VTA provides booking services for rental cars by combining the services of several worldwide operating car rental agencies. Then, another VTA uses these services for providing an end-user service for booking complete holiday trips worldwide.

We provide the modeling of one such VTA use case in Section [3.1.VTA for International Online Train Ticket](#).

### 2.1.3 Actors, Roles and Goals

In the general use case there are 3 actors. The following defines why they participate in this use case (goal) and the particular interactions they are involved in (roles).

1. **Customer:** the end-user that requests a service provided by the VTA
  - *Goal:* automated resolution of the request by a user-friendly tourism service
  - *Role:* end-user, interacts with VTA for service usage, payment, and non-computational assets (e.g. receiving the actual ticket when booking a trip)
2. **Tourism Service Providers:** a commercial companies that provides specific tourism services
  - *Goal:* sell service to end customers, maximize profit as a commercial company
  - *Role:* provides tourism service as a Web Service (also provides the necessary semantic descriptions of the Web Services), has a usage and allowance contract with the VTA

3. **VTA**: the intermediate between the Customer and the Tourism Service Providers. It provides high-quality tourism services to customers by aggregating the separate services provided by the single Service Providers.
  - *Goal*: provide high-quality end-user tourism services, uses existing tourism services and aggregates these into new services, maximize profit as a commercial company / represent union of service providers (depending on the owners of the VTA).
  - *Role*: interacting with customer via user interface (can be web-based for direct human customers interaction or an API for machine-users), usage and allowance contract for Web Services offered by Service Providers, centrally holding all functionalities for handling Semantic Web Services (mechanisms for discovery, composition, execution, etc.)

## 2.1.4 Usage Scenarios

We identify the following usage scenarios

1. *VTA interacts with Service Providers on contract and Web Service usage and allowance*
  - **Participating Actors**: VTA and Service Providers
  - **Activities**: business contract negotiation
  - **Technological Requirements**: contract information requirements are modeled in the system, i.e. Web Service usage is implemented via Policies
  - **Possible Extensions**: contract negotiation can be supported by automated mechanisms
2. *Customer requests VTA for searching tourism service offers, VTA detects and queries suitable Web Services and displays results to Customer*
  - **Participating Actors**: Customer, VTA, Tourism Service Providers
  - **Activities**:
    - (1) Customer selects "Search" services as provided by the VTA
    - (2) VTA discovers, invokes and executes corresponding Web Services
  - **Technological Requirements**:
    - (1) VTA has to pre-define the "Search" functionality that can be requested by a Customer
    - (2) the Tourism Service Providers' Web Services must be semantically described in order to support dynamic discovery (assuming that single Web Services can perform the search functionality)
    - (3) VTA has to provide mechanisms for automated Service Discovery
  - **Possible Extensions**:
    - the Customer specifies its request in natural language and the request is translated into machine readable form and processed by the VTA service automatically
    - several Web Services are aggregated for providing the requested functionality if it cannot be fulfilled by a single service
3. *Customer selects a concrete offer and requests booking for this offer (interacting with the VTA), VTA detects and aggregates Web Services for booking (incl. booking, payment, etc.), displays result to Customer and handles complete execution of customer-interaction (computational part)*
  - **Participating Actors**: Customer, VTA, Tourism Service Providers
  - **Activities**:
    - (1) Customer selects one concrete offer out of the Search results of usage scenario 2
    - (2) VTA discovers and composes available Web Services from Service Providers and composes them into the functionality to satisfy the user request

(3) VTA executes the Web Services in the sequence determined, controls the execution (handles errors and detects alternative paths if a Web Service fails)

(4) VTA interacts with Customer during execution when further information is needed (e.g. a credit card number for payment)

- **Technological Requirements:** contract information information requirements are modeled in the system, i.e. Web Service usage is implemented via Policies

(1) Web Services must be semantically described in order to support dynamic discovery, composition, and execution

(2) VTA has to hold mechanisms for automated Service Discovery, Composition, and Execution

(3) VTA has to provide and interaction interface for contingent Customer-interaction during Service execution

- **Possible Extensions:** advanced mechanisms for automated execution of aggregated Web Services

4. *VTA interacts with Customer and Service Provider for non-computational parts (e.g. delivery of actual tickets)*

- **Participating Actors:** Customer, VTA, Tourism Service Providers

- **Activities:** customer notification, accounting, good delivery (out of computational system), etc.

- **Technological Requirements:** mechanisms for notification and accounting

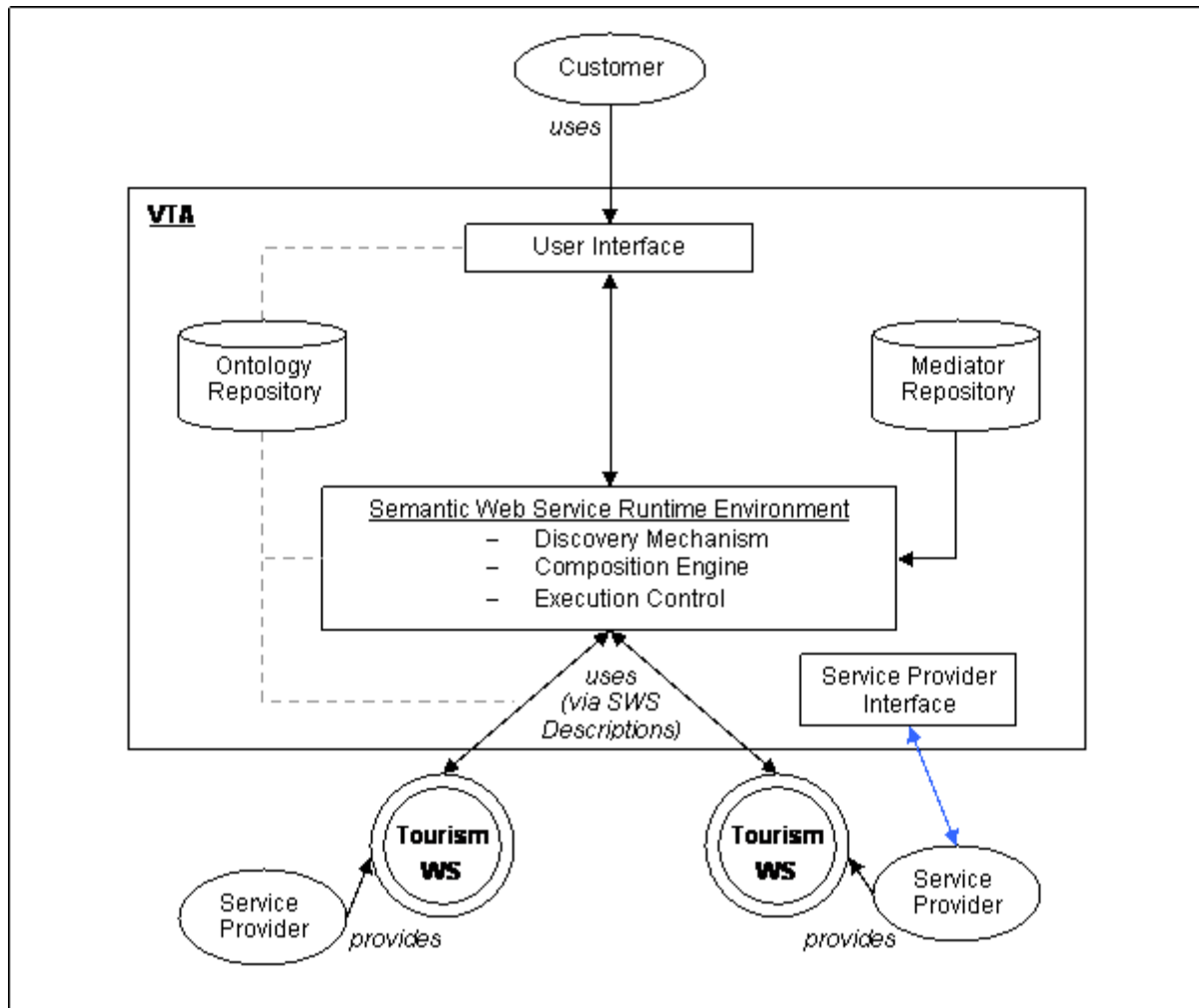
- **Possible Extensions:** Web Services can be used for:

- customer notification
- VTA-Service Provider interaction on accounting and good delivery mandate

## 2.1.5 System Architecture

In this use case, the VTA is the central point of interaction between the Customer and other Web Services. Regarding the technological requirements, it is obvious from the usage scenario descriptions that (1) the Web Services offered by the Service Providers have to carry sufficient descriptive information to support automated Web Service usage, and (2) that the VTA has to provide all mechanisms to handle Semantic Web Services. The basic architecture of such a VTA as a central entity for Semantic Web Services handling is shown in Figure 3. The essential functionalities of Semantic Web Service enabled VTAs – with special regard to the requirements for Semantic Web Service technologies – are:

- It has to provide a user interface for customer interaction (for both human and machine users)
- It has to discover suitable Web Services for an “instantiated” user request
- It has to invoke and combine external Semantic Web Services
- It has to provide a Web Service Execution Environment with control functions, error handling, and support for optional user interaction
- It has to have to deal properly with heterogeneous resources, thus allowing for appropriate mediation facilities.
- It has to provide interfaces for cooperation with Service Providers.



**Figure 3. General Architecture of a SWS-enabled VTA**

Summarizing, the VTA is a SWS-enabled B2C application that provides an end-user service following a client/server model. In order to support coherent functionality of the VTA and to ensure that the descriptions of Web Services are compatible to this, an overall framework for SWS technologies is needed. This is provided by WSMO.

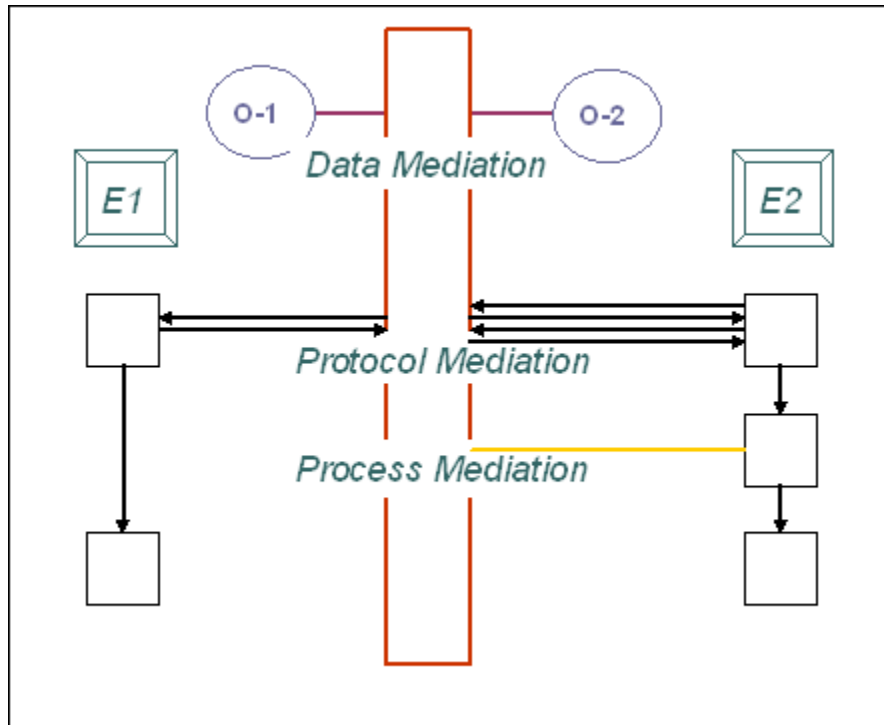
## 2.2 B2B - Integration with Semantic Web Services

The second use case is concerned with the integration of possible heterogeneous resources in B2B settings which is considered as one of the most important application fields of the Web Service technology.

### 2.2.1 Description

In the B2B use case, two enterprises called E1 and E2 want electronically exchange business documents across the network. Assuming that partners may not know each other before, contract negotiation and contract agreement are essential aspects of this use case. The contract agreement defines roles of enterprises in the conversation, for instance, one of the enterprise E1 becomes the seller and the second enterprise E2 becomes the buyer. An agreement also predefines the order of the messages interchanged between the parties, e.g. the buyer first sends purchase order (PO) and after that it receives purchase order acknowledgement (POA). In contrast to the previous B2C use case, where the client/server

model of interactions has been adopted, here partners are equal in the interaction, i.e. a peer-to-peer model is assumed in this use case. Each of the companies has an own set of web services for exchanging business documents electronically. The infrastructure provided by SWS takes care for any necessary mediation between web services (links web services), ontologies (resolves possible representation mismatches between ontologies used by these two enterprises), goals (links goals), as well as linking web services and goals. The infrastructure also supports the execution of the contract to fulfill approved agreement.



**Figure 4. B2B Integration with Semantic Web Services**

In this use case an ultimate goal of an enterprise E1 is to integrate its own back-end system with the back-end system of an enterprise E2. Once integrated, SWS software enables back-end systems of both companies to interact and to preserve the message, process and protocol semantic. The information systems used by enterprises E1 and E2 are **autonomous**, **heterogenous** and **distributed**. Semantic Web Services address each of these three properties and the software based on SWS enables companies to cooperate.

- The back-end systems in E1 and E2 are **autonomous** since each of them changes its state without informing other system about it. SMS software enables to track state changes of back-end applications to facilitate coordination between systems of E1 and E2.
- The back-end systems in E1 and E2 are **heterogenous**, because each of them has different conceptual model for expressing business semantics. SWS software takes care of appropriate mediation of the representation and meaning of the back-end system to the equivalent representation and meaning of the other system. The SWS software ensures to maintain the same semantics between back-end systems of E1 and E2.
- The back-end systems in E1 and E2 are **distributed** because each of them maintains its own state independently from the other system. Back-end applications in companies E1 and E2 do not share data or state at all. SWS software implemented in both companies takes care of transporting data between the systems.

## 2.2.2 Scope

The use case assumes peer-to-peer relationships between two business partners carrying conversation about purchasing/selling of goods. The B2B use case focuses on the technical infrastructure based on the SWS technology, which enable any business company to automatically discover web services which are capable to fulfill its goals, compose simple web services into complex web services to achieve a given goal and to automatically execute given services in a particular order. This use case assumes that there may be no prior business relationships between two enterprises before the discovery. Enterprise E1 must find enterprise E2 and they must agree and enforce the contract in their companies. Agreement should define roles of each of them in the agreed business process – e.g. one of them would become a buyer and one of them would become a seller. The agreement can lead to only one time execution of the agreed business process (e.g. request purchase order) or to long time relationships based on the multiply execution of the agreed contract. Payments are sent through financial institutions and at this stage they are out of the scope of this use case. The same situation concerns the shipment of the goods. This use case consider sending documents as for example purchase orders or invoices, but the physical shipment of goods is out of the scope of this use case.

## 2.2.3 Actors, Roles and Goals

There are two actors in the B2B use case – actors, which represent two business entities. The size and the importance of companies are not predefined in this use case. They might differ in size but from the perspective of this use case it should not matter which one of them is a more dominant partner. Both of the enterprises undertake a predefined role in the use case. These are:

1. **Buyer:** the company, which initiates the use case by searching for a partner, which is capable to sell goods.
  - *Goal:* Finding a business partner who is capable to provide goods. Signing the contract, discovering capabilities of the seller, composing provided web services and executing them.
  - *Role:* A business entity, which seeks business partner to achieve given goal by establishing new business relationships. Once the contract is signed it must be executed and as the result of contract execution, the buyer should receive goods. Buyer initiates the process described in this use case.
2. **Seller** - seller provides goods. It waits for buyers, responds to their requests, signs the contract and ships goods.
  - *Goal:* Providing goods. Signing the contract, discovering capabilities of the buyer, composing provided web services and executing them.
  - *Role:* A business entity, which waits for the partner to establish business relationships. As the result of the execution of the contract, the seller should send goods the seller.

## 2.2.4 Usage Scenarios

In this use case the following usage scenarios have been identified:

1. *Contract negotiation and implementation of agreement between buyer and seller.*
  - **Participating actors** - buyer and seller
  - **Activities** - business contract negotiation and implementation
  - **Technological Requirements** - The technology should enable matching goals of a

buyer with capabilities of a seller. But matching goals of capabilities is not sufficient, because once goal is matched with the capability, the interfaces of two businesses should be matched as well.

- **Possible Extensions** -contract is negotiated and implemented completely automatically by appropriate infrastructure

2. *Typical business messages exchange (e.g. PO & POA exchange);*

- **Participating actors** - buyer and seller

- **Activities** - buyer sends PO to seller. Buyer can at any time check the status of processed order. Seller sends back POA. Lower level acknowledgments messages for each of the PO and POA can be also exchanged.

- **Technological Requirements** - The technology should enable conversation between business partners, supporting different process models to achieve given task e.g. buying a product. For example system of one business partner might require a synchronized confirmation for each business document send out, while the system of the other business partner assumes that once the document is send, it does not have to be confirmed. The SWS platform should provide appropriate process mediation mechanism to resolve this issue.

- **Possible Extensions** - the system of one of the business partner might failed and drop in the middle of conversation (e.g. it receives PO, but never sends a POA). The SWS platform, similarly to workflow engines, takes care to recover from deadlock and livelock errors.

3. *SWS infrastructure crashes - once it recovers, it reliable commence its operations*

- **Participating actors** - buyer and seller

- **Activities** - Because of some internal (e.g. lack of power supply for the server) or external (e.g. lack of network connection) failure, the SWS system becomes temporary unavailable. Once it is back online, it commence from the point where the execution has been dropped. None of the messages are lost, none of the processes are executed from the beginning.

- **Technological Requirements** - Reliable and event driven architecture.

- **Possible Extensions** - The SWS infrastructure informs all interested parties that it is back online.

4. *E1 and E2 want to deploy a new integration definition type (described in WSML). The developer responsible for the SWS software writes a new integration type, which is next deployed by SWS infrastructures in both enterprises.*

- **Participating actors** - buyer and seller

- **Activities** - Any new integration type can be compiled and deployed by SWS infrastructure.

- **Technological Requirements** - Standard interface which allows carrying conversation between SWS infrastructure and WSML editor. New integration definition types can be saved and retrieved from the system.

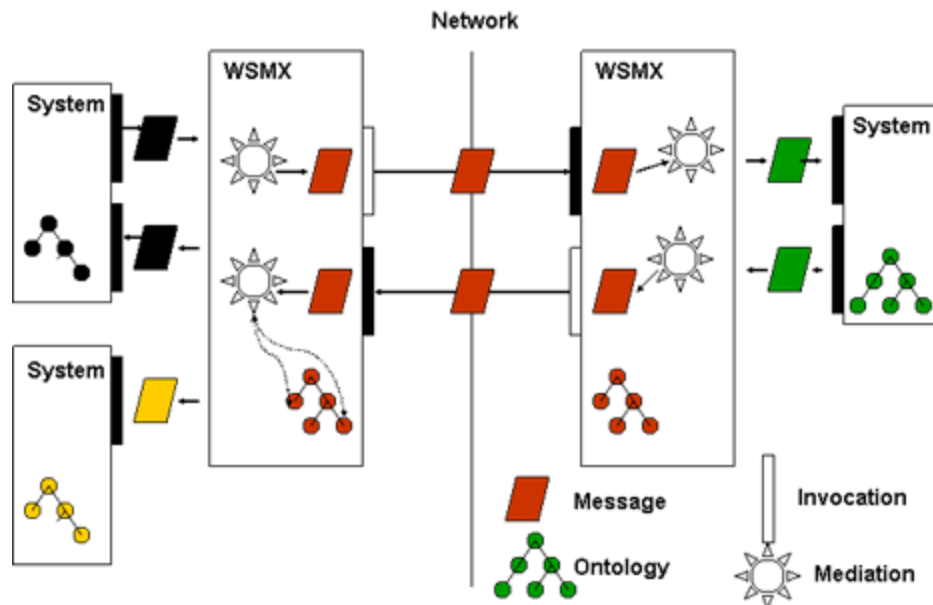
- **Possible Extensions** - Public interface which enables any external party to provide own definitions.

## 2.2.5 System Architecture

The Web Services Modeling Execution (WSMX) is the infrastructure is a WSMO-reference implementation that addresses this use case, see WSMX homepage at:

<http://www.wsmo.org/wsmx/>. Therein, a WSMX-platform is hosted by each of the enterprises to support services following a peer-to-peer model. WSMX is software implementation of a web service execution environment supporting the development, management and execution

of Semantic Web enabled Web Services. WSMX platform does not differentiate between calls coming from the back-end application systems (intra-company information systems) and from the information systems of other enterprises. WSMX can also communicate directly with other WSMX platforms hosted by other enterprises as shown on figure 5.



**Figure 5. B2B Use Case System Architecture**

### 3. WSMO Use Cases

This section gathers different use cases developed around WSMO, each with a different focus. We briefly introduce the use case here, while the use case modeling is provided in a different document.

#### 3.1. B2C - a Virtual Travel Agency for Online Train Tickets

This use case models a B2C application scenario: a Virtual Travel Agency for purchasing train tickets provides a WSMO Web Service.

This is the first WSMO use case developed in previous versions of this document.

Link: <http://www.wsmo.org/2004/d3/d3.3/>

#### 3.2. B2B - Business Document Exchange

This use case models a B2B scenario for exchanging business documents, this scenario will be more elaborated in future, for now there is only a placeholder document:

Link: <http://www.wsmo.org/2004/d3/d3.4/>

#### 3.3. Semantic Web Fred - Agent Collaboration with Semantic Web Services

This is the Use Case defined for Semantic Web Fred - an agent system for automated, cooperative goal resolution that realizes WSMO. More information on the SWF project can be found at the SWF project website at: <http://www.deri.at/research/projects/swf/>.

Link: <http://www.wsmo.org/2004/d3/d3.5/>

## 4. Conclusions and Further Work

The aim of this document is to identify the challenges for Semantic Web Services by elaborating usage scenarios, and to gather use cases that cover specific aspects around Semantic Web Service modeling and technologies around WSMO, serving as a testbed for development of WSMO-based technologies. The most important outcomes of this deliverable are:

- identification of aims and challenges for Semantic Web Service technologies
- gain a better understanding of WSMO, its distinct components, and how these are related
- development of WSMO by recursion and application scenario testing.

This deliverable is intended to evolve over time. The directions for future work in this deliverable are:

- specify additional use cases that will address certain aspects of Semantic Web Services
- test and recursive development of WSMO according to results in use cases.

## References

**[Arroyo et al., 2004]** Arroyo, S.; Lara, R.; Gómez, J.; Berka, D.; Ding, Y.; Fensel, D. (2004): Semantic Aspects of Web Services. In Munindar. P. Singh (Ed.), Practical Handbook of Internet Computing. Baton Rouge: Chapman Hall and CRC Press, Baton Rouge. 2004.

**[Arroyo and Stollberg, 2004]** Arroyo, S.; Stollberg, M.: *WSMO Primer*. WSMO Deliverable D3.1, available at: <http://www.wsmo.org/2004/d3/d3.1/v0.1/>

**[ebXML]** ebXML, available at: <http://www.ebxml.org>.

**[EDIFACT]** UN/EDIFACT, available at: <http://www.unece.org/trade/untdid/welcome.htm>.

**[Fensel & Bussler, 2002]** D. Fensel and C. Bussler: *The Web Service Modeling Framework WSMF*, Electronic Commerce Research and Applications, 1(2), 2002.

**[He et al., 2004]** he, H.; Haas, H.; Orchard, D.: *Web Services Architecture Usage Scenarios*, W3C Working Group Note 11 February 2004. available at: <http://www.w3.org/TR/ws-arch-scenarios/>.

**[Hayes (Ed.), 2004]** P. Hayes: *RDF Semantics*, W3C Recommendation 10 February 2004, 2004.

**[Roman et al., 2004]** D. Roman, U. Keller, H. Lausen (eds.): *Web Service Modeling Ontology - Standard (WSMO - Standard)*, version 0.2 available at <http://www.wsmo.org/2004/d2/v02/>.

**[SOAP]** Mitra, N.: *SOAP Version 1.2 Part 0: Primer*. W3C Recommendation 24 June 2003. available at: <http://www.w3.org/TR/soap12-part0/>

**[UDDI]** Bellwood, T.; Clément, L.; von Riegen, C. (Ed.): *UDDI Version 3.0.1*. UDDI Spec Technical Committee Specification, Dated 20031014. available at: [http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm)

**[WSDL]** Chinnici, R.; Gudgin, M.; Moreaum, J.-J.; Weerawarana, S. (2003): *Web Services Description Language (WSDL) Version 1.2*. W3C Working Draft 3 March 2003. available at <http://www.w3.org/TR/wsdl20/>.

## Acknowledgements

The work is funded by the European Commission under the projects [DIP](#), [Knowledge Web](#), [SEKT](#), [SWWS](#), [Esperonto](#), and [h-TechSight](#); by [Science Foundation Ireland](#) under the [DERI-Lion](#) project; and by the Vienna city government under the [CoOperate program](#).

The editors would like to thank to all the [members of the WSMO working group](#) for their advice and input into this document.

---

## Appendix: Change Tracking

To facilitate retracing of changes inbetween different version of this deliverable, the following lists the essential changes done in comparison to the preceding version.

The change tracking starts with the version of 28 June 2004.

**Version: 08 October 2004** <http://www.wsmo.org/2004/d3/d3.2/v0.1/20041008/>

- updated links to reflect new split of deliverables

**Version: 04 October 2004** <http://www.wsmo.org/2004/d3/d3.2/20041004/>

- changed structure of deliverable: this is an overview document, while the actual use cases are provided in separate documents  
- adopted B2C Use Case to WSMO Standard version 1.0

**Version: 19 July 2004** <http://www.wsmo.org/2004/d3/d3.2/v0.1/20040719/>

- ontologies: rationales and updates, PO Ontology currently under development  
- added general Goal and GG Mediator; the concrete Goal is derived from these  
- updated WS Capability (assumption is now that the credit card is valid)

**Version: 28 June 2004** <http://www.wsmo.org/2004/d3/d3.2/v0.1/20040628/>

- complete read-thru with corrections of deliverable text (regarding comments from Jos de Bruijn)

- corrections of domain ontologies
    - \* changed section 3.1.1 to "Use Case Overview", describes the properties of the WSMO components modeled below
    - \* the web service described now is understood as an aggregated / composed web service that offers the overall functionality for purchasing train tickets online. In later versions, the Choreography description as well as the Orchestration with specific Web Services for searching and buying train tickets can be adopted.
    - \* corrected / clarified descriptions for modeling descriptions.
  - correction of WSML-models for Goals, Web Services, Mediators
  - revised the Web Service Discovery description (section 3.1.3)
  - updated the FLORA2 resources to the WSML models (as in Listings)
  - namespace handling refined
- 



\$Date: 2004/10/08 13:38:51 \$

[webmaster](#)