



D12v0.1. Web Service Modeling Ontology - Full (WSMO - Full)

WSMO Working Draft 24 June 2004

This version:

<http://www.wsmo.org/2004/d12/v0.1/20040624/>

Latest version:

<http://www.wsmo.org/2004/d12/v0.1/>

Previous version:

<http://www.wsmo.org/2004/d12/v0.1/20040404/>

Editors:

Chris Preist
Dumitru Roman

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.

Table of contents

- [1. Introduction](#)
 - [1.1 Aims and purpose of WSMO-Full](#)
 - [1.2 Approach](#)
 - [2. What is a service?](#)
 - [3. Conceptual model of WSMO-Full](#)
 - [3.1 Discovery in WSMO-Full](#)
 - [3.2 Service definition in WSMO-Full](#)
 - [3.3 Service execution in WSMO-Full](#)
 - [4. Conclusions and further work](#)
 - [5. Glossary](#)
 - [References](#)
 - [Acknowledgement](#)
-

1. Introduction

WSMO-Full aims to extend the conceptual model of WSMO-Standard [[Roman et al., 2004](#)] (which describes minimal characteristics that should be common to all services accessible on the web) by bringing a B2B perspective to it.

1.1 Aims and purpose of WSMO-Full

WSMO-Full aims to provide a conceptual model and a context for understanding semantic web services and the relationships between the parts this model consists of.

This deliverable is intended to provide a common definition of a semantic web service based on the way a web service is described in WSMO-Standard, extended by elements required in a real B2B scenario, and define its place within a larger semantic web services framework. WSMO-Full, as an extension to WSMO-Standard, aims to be an "interoperability" conceptual model: it defines the global elements of the global semantic web services network that are required in order to ensure interoperability between semantic web services.

It is not the aim of the WSMO-Full to specify how semantic web services are implemented, and imposes no restriction on how these services might be combined.

1.2 Approach

WSMO-Full adopts the view that a service is centered around the **contract** between the service provider (the entity which provides the service) and service requestor (the entity which request the service). Such a contract can be explicit or implicit, and can be agreed in a variety of ways. It is important to note that the contract represents an agreement about what a service does; it does not necessarily need to have any legal status.

The potential role of contracts in supporting web service interactions is becoming increasingly recognised. The importance of contracts in supporting interactions between agents in open systems, especially e-commerce, has long been recognised by the agent community (For example, [\[Jennings et al., 1996\]](#)). Furthermore, proposals have been made to use contracts in semantic web services [\[Grosf & Poon, 2004\]](#).

The document is further structured in the following way: [Section 2](#) gives a definition of what a service is, [Section 3](#) gives an overview of the WSMO-Full conceptual model, including the approach to discovery, service definition and service execution. [Section 4](#) presents the conclusions and further intentions and [Section 5](#) contains a glossary, which consists of relevant terminology used in this document.

2. What is a service?

The word 'service' can be used in several different ways. If a clear conceptual model is to be developed for semantic web services, we believe that it is necessary to make these different uses explicit.

- 'Service' as provision of value in some domain.

In the economy nowadays, a great variety of different services are traded; cleaning of windows, provision of stock quotes, shipment of crates, showing of films, giving massages, translation of documents, location of out-of-print books, provision of a broadband connection of a certain quality etc. In each case, the customer receives something of value to them in return for a payment to the supplier. The service the customer receives can be described in some language about the domain of interest. Such a description does not need to refer to the

way in which the supplier and customer and interact. This is what an economist or businessperson would refer to as a service.

On the internet, services can be provided which involve some financial transaction, and so correspond closely to the above perspective. However we can broaden this view of service without losing what is essential. A service can be provided without any payment, and the value the service provides need only be 'in the eyes of the requestor', not necessarily something that would be considered to have monetary value in the commercial world. For example, a service may be the provision of a specific piece of information from a database, the provision of a plan to achieve some goal, or the assignment of some domain task to a given software application. Hence, we define a service to be the provision of something of value, in the context of some domain of application, by one party to another.

We also need to distinguish between a particular provision of value (for example, location of a particular out-of-print book) from the general capability to provide (for example, the ability to locate out of print books). Here, we borrow loosely from object oriented language. We refer to the former as a concrete service, and the latter as an abstract service. Hence an abstract service is defined as the capacity to perform something of value, in the context of some domain of application. This is close to the definition of service given in the [WSA](#).

- 'Service' as a software entity able to provide something of value.

This usage is common in the computer science and IT community. Papers often speak of sending messages to services, receiving results from services, executing services etc. While this is clearly appropriate in the context of web services and service oriented architectures, it can cause some confusion when mixing usage of this definition with the definitions above. For that reason, we believe that it is more accurate to speak of such an entity as being (part of) a service provider agent (SPA). We also use the term 'web service' to refer to a specific application with a WSDL/SOAP interface.

- 'Service' as a means of interacting online with a service provider.

The most common example of this usage is 'negotiation service'; A negotiation service provides the means to interact (negotiate) with a given actor. This usage is clearly different from the first usage, in that negotiation in itself does not provide something of value. However, the outcome of the negotiation may do so (and hence may be a service in the first sense.) For this reason, we refer to this as the provision of a negotiation 'protocol' or 'choreography'.

All three of these aspects of service provision are essential; the 'value' in some domain, the interactions involved, and the implementation of the entity able to deliver what is needed. However, we believe that for conceptual reasons it is best to make clear distinctions between them.

3. Conceptual model of WSMO-Full

The conceptual model of WSMO-Full is based on the distinction between the *abstract service* a service provider offers and the *agreed service* which appears within a contract, and assigning semantics to each. Additionally, we introduce the *concrete service* which is the performance of a specific service at a particular time. Similarly,

we consider *abstract tasks*, *agreed tasks* and *concrete tasks*. Firstly, we define the concept of a concrete service. This is the core 'building block' with which to define the semantics of more abstract descriptions of services.

A **Concrete Service** is an actual or possible performance of a set of tasks that represent a coherent functionality (and therefore deliver some value) within some domain of interest to its associated service requestor and service provider.

A concrete service:

- has a set of one or more concrete tasks
- has a set of messages
- has a service provider and service requestor
- has one or more post-agreement choreographies associated with it
- has a service provider interface and/or a service requestor interface which are used by the choreographies
- may be a resource

A concrete service is effectively a (possible or actual) specific and detailed set of tasks carried out at a given time, and the messages exchanged in association with this. The messages are exchanged between a specific provider and requestor, according to certain choreographies and using certain interfaces.

An example of a concrete service would be: The service provider PayneConverter performs the task of quoting a euro exchange value for £10 for the service requestor cwp at 2pm on May 25, 2004. The message exchange consists of the requestor sending the message exchangeRateRequest(10,uks,euro) and the provider replying with exchangeRateResponse(14.9599, 2004.05.25 14:00:29 GMT).

A service provider, however, does not offer a specific concrete service. Rather, they are able to provide many different possible concrete services. Similarly, a requestor is rarely after a specific concrete service; they are interested in many possible services which meet their needs. For that reason, we define the concept of abstract service.

An **Abstract Service** is some set of concrete services. A concrete service is said to be a *realization* of an abstract service if it appears in this set.

An **Abstract Service Description** is some machine-processable description which has, as its semantics, an abstract service. An Abstract-Service Description is some specification of a set of concrete services. The format of this description is not specified as part of WSMO-Full. There are several possibilities:

- An extensional listing of concrete services.
- A label representing the name of some abstract service in some ontology of services.
- A template or formal description in some knowledge representation language.

Option one is the most conceptually straightforward, but in practice will be infeasible except in the simplest of cases. In option two, the semantics of the label is defined as the (possibly infinite) set of concrete services which correspond to that label within the ontology. This correspondence may be well-defined and explicit, with a service specification or contract template associated with the label. Often, however, the correspondence will be more informal. For example, a label 'bookselling' may have as its semantics the set containing any possible concrete services where books (and only books) are being sold. The third option specifies the set of concrete services by

explicitly and formally defining an abstract service using some knowledge representation (KR) language such as OWL or F-logic. This may involve explicit descriptions of abstract tasks, possibly including parameters. The set-theoretic semantics of the KR language then defines the set of concrete services to be those which formally satisfy the Abstract-Service Description. Such a representation may leave the choreographies to deliver the tasks undefined. In such a case, the abstract service will include all possible choreographies which can be associated with those tasks. Alternatively, the abstract service may place restrictions on the choreography (e.g. it may be constrained to be a RosettaNet PIP) or may define a specific required choreography.

A resource representing an abstract service will have an Abstract-Service Description associated with it. In addition, it:

- may have a contract agreement choreography associated with it
- may have a service provider, and an associated service provider interface
- may have a service requestor, and an associated service requestor interface

Whereas a concrete service is always associated with a specific requestor and provider, an abstract service may have one or both of these undefined. Similarly, an abstract task may be less completely defined than a concrete task is.

An abstract-service description provides a unifying view on both service provider and service requester. However, as mentioned above, an abstract service may have only one of service requester or service provider defined: when only the provider is defined then the abstract-service description will correspond to **WSMO-Standard service description** and when only the requestor is specified then abstract-service description corresponds to a **WSMO-Standard goal description**.

An example of an abstract service would be: The service provider PayneConverter can perform the task of quoting an exchange value for any amount, between any two currencies, at any time in the future. The abstract service is the set of all concrete services which meet this description. (Furthermore, a formalized version of this would be an example of an Abstract-Service Description.)

An abstract service, as we shall see below, can be used for advertising or requesting. However, it is not precise enough to define what a requestor and provider agree. For that we define the following:

An **Agreed Service** is an abstract service agreed between two parties. Hence it is associated with a specific service provider and service requestor, together with their associated interfaces. It also has specific post-agreement choreographies associated with it. Its definition is such that:

- Any further decisions necessary to select a realization will occur as choices made during a post-agreement choreography.
- Any choice made by one party during the post-agreement choreography is such that the other party is indifferent to how that choice is made.

An example of an agreed service would be: The service provider PayneConverter agrees to perform the task of quoting a euro exchange value for any amount of UK sterling for the service requestor cwp any time before 12am on May 26, 2004. PayneConverter is indifferent as to the exact value it converts and the time of conversion. Its choreography definition puts the decision of how much (via a parameter) and when (via choice of when to send the message) in the hands of cwp.

Hence this is indeed a possible agreed service.

The agreed service forms the core of the service contract between two parties, which defines the semantics of their interaction:

A **Service Contract** is an agreement between a service provider and requestor that the provider will supply an agreed service to the requestor. The contract:

- has a service provider
- has a service requestor
- has an agreed service description
- has an association between the tasks involved in the agreed service description and choreographies and message interfaces which will be used to have conversations about the tasks.
- may have information about who is responsible and what compensation may occur in the case that a service is not delivered correctly. Compensation may involve remedial actions and/or payment of damages.
- may have additional legal terms and conditions.

The **agreed-service description** will be (explicitly or implicitly) in terms of the tasks to be performed. It may include tasks to be performed by the requestor (e.g. payment transfer in the case of e-commerce applications.) The task/choreography association defines how the interaction between the parties can take place during delivery of the concrete service. This may be done by including the choreographies explicitly in the contract. This is appropriate where a choreography may vary and require agreement between two parties. Alternatively, it may be done by referencing a choreography definition elsewhere. This is appropriate when the service provider agent offers a specific choreography and is not able or willing to alter it, or provides a choice between a small number of choreographies, or the parties use some standard choreography defined elsewhere.

In the following sections we present the conceptual model of WSMO-Full. Firstly, we present definitions of the concepts we introduce. Then we present and explain the conceptual model. For clarity, we separate it into three diagrams, representing the entities and relationships involved in discovery, service definition and service execution. We omit the entities and relationships which appear in the WSA if they remain unchanged in our model and are not relevant to the discussion of service semantics. We adopt the notational conventions used by the WSA; in particular, we provide textual descriptions and diagrammatic representations of concepts and relationships between these concepts. Many of the relationships are descriptive, but 'has' is treated as definitorial. In the diagrams, boxes represent concepts and arrows represent relationships. The arrow points from the subject of the relationship to the object.

3.1 Discovery in WSMO-Full

[Figure 1](#) illustrates the concepts involved in discovery. A requestor agent requires one of a class of abstract service; to this end, they publish a description of their requirements as a WSMO Goal. Similarly, a provider agent is able to provide services within a certain class of abstract service. It publishes a description of the services it can provide as a WSMO service description.

Ideally, in each case, the description of the abstract service will correspond exactly to the actual abstract service which is required or offered. However, in practice, this may

not be possible.

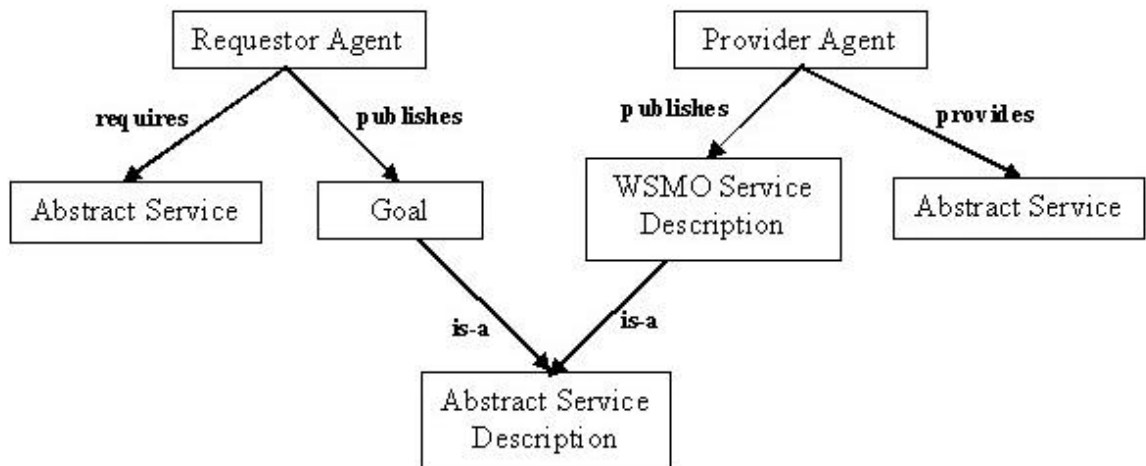


Figure 1. Discovery in WSMO-Full.

3.2 Service definition in WSMO-Full

Following discovery, two parties need to determine if they can agree a service which the provider will supply to the requestor. This is done through the contract agreement phase. In some cases, this phase will be unnecessary and the results from the discovery phase will be enough. This is the case if the following criteria are satisfied:

- The offer description provides an explicit service description which includes post-agreement choreographies.
- Any aspect of the offer description which has variability (for example, a parameter which is not defined) is such that the provider is indifferent with regard to the choice.
- The choreography used to deliver the service is structured such that choices within this variability (such as a parameter instantiation) can be input through it.

If all three of these criteria hold, then the contract agreement phase is unnecessary. The service offer description (WSMO service description) defines a set of concrete services, and the provider doesn't care which is selected. The delivery choreography allows the selection of a concrete service to be made by inputting parameter values chosen by the requestor.

If the published abstract service description does not satisfy the criteria, but the service provider agent is able to give an abstract service description which does, then the contract agreement phase becomes a straightforward 'take-it-or-leave-it' process. The contract agreement choreography simply consists of the requestor accessing the description. It then goes on to execute the delivery choreography if it chooses to use this service provider, and does nothing otherwise.

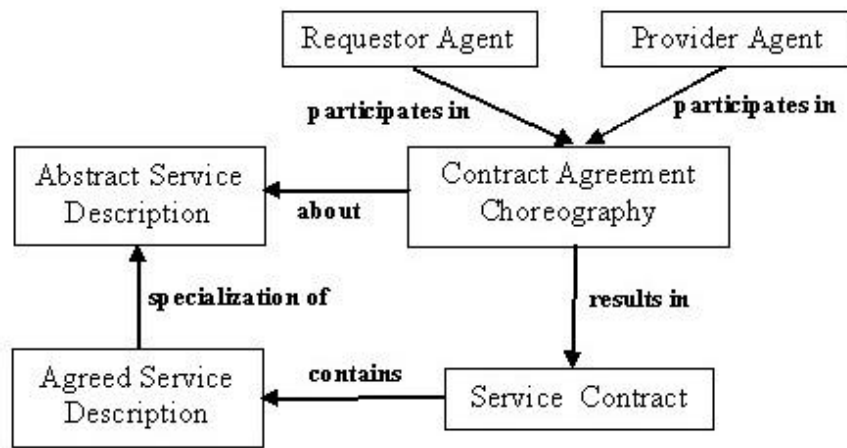


Figure 2. Contract Agreement Model of WSMO-Full.

In general, the contract agreement phase can be more complex than this. [Figure 2](#) illustrates the concepts it involves. At least one provider and at least one requestor agent participate in a choreography to agree a contract between them. The choreography involves discussing some abstract service, and may explicitly involve the manipulation of a description of that abstract service. If the choreography terminates successfully, it results in a service contract agreed between two parties; this contract contains a description of an agreed service which specifies the concrete service to be delivered. The agreed service will be a specialization of the abstract service found during discovery. The decision making within the contract agreement phase has three aspects:

- Partner selection - Often, one or both parties have a choice about which partner to select to deliver/receive a service. Specifically, it is often the case that a service requestor has a choice of several potential providers. Similarly, a service provider that has limited resources and is in demand may need to make a choice to supply some subset of requestors currently demanding its services.
- Service selection - One party may need to select and specify certain parameters of a service which weren't specified during discovery. Usually, this is the requestor making their requirements more specific (e.g. specifying exactly which books they wish to purchase from an online book-seller.)
- Negotiation- Often, when a service specification is fully defined, other parameters of a contract such as price will be fixed. However, in some circumstances, they may be open to negotiation. This means a previously proposed price (or other parameter) may be relaxed by one party in an effort to reach an agreement.

Any given contract agreement phase does not necessarily involve all three of these. Furthermore, the order in which these take place is not fixed; they may take place one after another in any order, or all three aspects can be interleaved with each other. In the agent mediated electronic commerce literature, a clear distinction is made between a negotiation protocol, defining the rules of negotiation, and the negotiation algorithm executed by a given agent [[He et al., 2003](#)]. Similarly, we distinguish here between a contract agreement choreography (in which two or more agents participate) and the algorithms within requestor and provider agents which they use to participate in such a choreography. The choreography is within the scope of the conceptual model, as it is inter-agent. The algorithms used by the agents, however, are private and so fall beyond the scope of the conceptual model.

A **contract agreement choreography** is a protocol for message interactions between

at least one service requestor and at least one service provider which, on successful termination, will result in the agreement of a service contract between two of the parties.

A contract agreement choreography may explicitly involve service selection and/or negotiation. Service selection protocols can take different forms. Examples include:

- A service provider may offer a list of all contracts they are willing to offer, and the requestor simply selects one.
- In the common 'shopping trolley' conversation, a requestor iteratively builds a definition of the service they wish to receive, and are given a price quote when they are satisfied. They can accept the quote, quit, or make alterations to the required service.
- 'Pseudo negotiation' may take place where partial descriptions of services (possibly using constraints on parameters) are exchanged between both parties in a series of proposals/counter-proposals. Slowly, by iterating, one party (usually the requestor) makes selections within the range and moves towards a fully instantiated contract. In this case, the provider has a set of fixed-price contracts but chooses to reveal these iteratively through a process which appears like negotiation. Sometimes this can be to make life easier for the requestor, as the provider co-operatively reveals contracts based on the revealed needs of the requestor. Sometimes, it can be in an attempt to guide the requestor to the provider's preferred options.

Example negotiation protocols include:

- Propose/counter-propose – Each party proposes a contract, and the other party either accepts it or proposes an alternative.
- Constraint relaxation – Parties exchange a set of constraints on a contract template. Agreement occurs when the union of the two sets is satisfiable. Until this point is reached, parties iteratively relax their set.
- Reverse Auction – A requestor sends out a call for quotes to fulfill a given service contract; providers respond with their cheapest quote, and iteratively place lower offers until no provider is willing to reduce any more. Agreement is made with the cheapest.

Many other possible approaches to negotiation exist ([\[He et al., 2003\]](#), [\[Vulkan, 2003\]](#)).

A choreography may also combine the two – for example, we could imagine a book selling service provider in which you use a shopping basket to choose your books, and then use propose/counterpropose to negotiate the final price. Usually, choreographies specifying these protocols will involve two parties (requestor and provider). However, either party may be involved in several simultaneous choreographies to allow them to make the optimal trading partner selection. Trading partner selection, therefore, is a consequence of the internal algorithm rather than the choreography. These parallel conversations may be hidden or may be deliberately made public as in the reverse auction. In the latter case, the conversation protocol can be viewed as part of a multi-party choreography. The definition of the choreography would be made public to all participants.

The association of a contract agreement choreography to a given set of provider and requestor agents must take place following discovery. The most common way to do this is for the advertising agent to associate a (pointer to) a contract agreement choreography description with their advert, or to return a contract agreement

choreography when they are first queried directly. The contract agreement choreography may take the results of the discovery phase as input, if the discovery phase revealed more information than just possible service providers. For example, the discovery phase may return a contract template for each service provider, and this can then be instantiated during the choreography. Alternatively, a provider agent may provide a new (and more detailed) Abstract-Service Description at the beginning of the choreography, or may allow a requestor to iteratively construct a service contract.

3.3 Service execution in WSMO-Full

When a contract agreement choreography has terminated successfully, then service delivery can take place. This may commence immediately on agreement of the contract, or may be initiated later by a subsequent choreography. [Figure 3](#) illustrates the concepts involved in the service delivery phase, most of which have already been defined. A concrete service is supplied by the provider agent to the requestor agent. This concrete service must be a member of the agreed service (which, recall, is a set of concrete services). The agreed service has an associated service contract which contains a description of the service agreed together with a mapping from the agreed tasks within the service to choreographies about these tasks. To distinguish these choreographies from the contract agreement choreographies, we refer to them as *post agreement choreographies*.

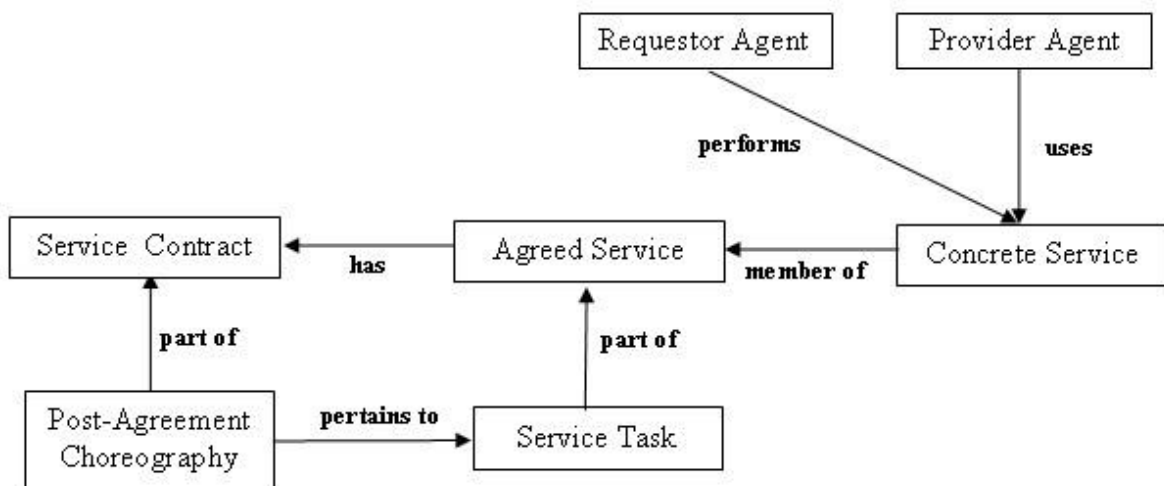


Figure 3. Service Delivery Model of WSMO-Full.

A **post-agreement choreography** is a protocol for a conversation between a service requestor and service provider about a contract which has been previously agreed, or one or more tasks within such a contract.

There are at least three different kinds of post-agreement choreography:

- **Delivery choreography** - This governs conversations which directly initiate and/or control the execution of one or more tasks in a contract. For example, a request/response to an information service provider returning the required information, or a payment protocol transferring money from one party to the other.
- **Monitoring choreography** - This does not directly control the execution of a task, but allows the requestor to monitor the task's progress.
- **Cancellation/renegotiation choreography** - This allows the cancellation or alteration of a service within a contract in certain circumstances. For example, a book seller such as Amazon may provide the option to cancel or alter an order

after it has been submitted, providing the order has not entered the dispatch process.

4. Conclusions and futher work

This draft presented a conceptual model for semantic web services developed out of a requirements analysis of a set of case studies [1] in the [SWWS project](#). Next drafts of this documents will include mediation in WSMO-Full and a more detailed alignment to WSMO-Standard.

This conceptual model will be further refined and developed based on different case studies which will be provided in the context of DIP, SEKT and KW projects.

5. Glossary

Service Provider	An entity that provides a service.
Service Requester	An entity that wishes to make use of a service.
Service Provider Agent	A provider agent is a software agent that is capable of and empowered to perform the actions associated with a service on behalf of its owner – the service provider.
Service Requester Agent	A requester agent is a software agent that wishes to interact with a service provider agent in order to request that a task be performed on behalf of its owner — the service requester.
Concrete Service	An actual or possible performance of a set of tasks that represent a coherent functionality (and therefore deliver some value) within some domain of interest to its associated service requestor and service provider.
Abstract Service	Some set of concrete services.
Agreed Service	An abstract service agreed between two parties.
Abstract Service - Description	Some machine-processable description which has, as its semantics, an abstract service.
Agreed Service - Description	Some formal or informal description of the agreed service's tasks to be performed.
Service Contract	An agreement between a service provider and requestor that the provider will supply an agreed service to the requestor.
Task	An action or combination of actions.
Resource	Anything that can have an identifier.
Contract-agreement choreography	A protocol for message interactions between at least one service requestor and at least one service provider which, on successful termination, will result in the agreement of a service contract between two of the parties.
Post-agreement choreography	A protocol for a conversation between a service requestor and service provider about a contract which has been previously agreed, or one or more tasks within such a contract.

References

[Grosf & Poon, 2004] Grosf, B. and Poon, T.: SweetDeal: Representing Agent Contracts With Exceptions using Semantic Web Rules, Ontologies and Process Descriptions. To appear, *International Journal of Electronic Commerce* (2004)

[He et al., 2003] He, M., Jennings, N.R. and Leung, H: *On Agent Mediated Electronic Commerce*. IEEE Transactions on Knowledge and Data Engineering 15(4) (2003) 985-1003

[Jennings et al., 1996] Jennings, N.R., Faratin, P., Johnson, M.J., O'Brien, P. and Wiegand, M.E.: *Using Intelligent Agents to Manage Business Processes*. Proceedings of the First Int. Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (1996) 345-360

[Roman et al., 2004] D. Roman, U. Keller, H. Lausen: *Web Service Modeling Ontology - Standard*, version 0.1 available at <http://www.wsmo.org/2004/d2/v0.2/>

[Trastour et al., 2001] Trastour, D., Bartolini, C. and Gonzalez-Castillo, J.: *A Semantic Web Approach to Service Description for Matchmaking of Services*. In Proceedings of the Semantic Web Working Symposium, Stanford, CA, USA, July 30 - August 1, 2001

[Vulkan, 2003] Vulkan, N.: *The Economics of E-Commerce*. Princetown University Press, Princetown, New Jersey (2003)

Acknowledgement

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.

[1] The SWWS use cases are available at <http://swws.semanticweb.org>

[webmaster](#)