



**HAL**  
open science

## Services Web : revue des approches de description sémantique

Nomane Ould Ahmed M'Barek, Samir Tata

► **To cite this version:**

Nomane Ould Ahmed M'Barek, Samir Tata. Services Web : revue des approches de description sémantique. SIIE 2008 : Système d'Information et Intelligence Economique, Feb 2008, Hammamet, Tunisie. pp.1 - 7. hal-01378724

**HAL Id: hal-01378724**

**<https://hal.science/hal-01378724>**

Submitted on 10 Oct 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Services Web : revue des approches de description sémantique

Nomane Ould Ahmed M'bareck  
*GET/INT, CNRS UMR SAMOVAR, Evry, France*  
*9 rue Charles Fourier 91011 Evry, France*  
*Nomane.Ould\_ahmed\_mbarek@int-edu.eu*

Samir Tata  
*GET/INT, CNRS UMR SAMOVAR, Evry, France*  
*9 rue Charles Fourier 91011 Evry, France*  
*Samir.Tata@int-edu.eu*

## ABSTRACT

Le Web est entrain de subir deux grandes évolutions à savoir les services Web et le Web sémantique. D'une part, l'évolution récente des technologies de l'Internet, guidée par le langage XML et ses technologies sous-jacentes, étend le rôle du Web d'un simple support d'information vers un intergiciel d'applications B2B. Cette nouvelle vague de l'Internet est guidée par les services Web. Les services Web sont des applications auto-descriptives, modulaires et faiblement couplées qui fournissent un modèle simple de programmation et de déploiement d'applications, basé sur des normes, et s'exécutant au travers de l'infrastructure Web [KAD 03]. La collaboration entre les services Web s'appuie sur une architecture à trois composants à savoir : le fournisseur, le client et l'annuaire.

Le Web sémantique, quant à lui, est une extension du Web actuel, dans lequel les informations ont un sens bien défini afin de permettre aux ordinateurs et aux utilisateurs de travailler en coopération [BUR 01].

L'application du Web sémantique aux services Web a donné naissance aux services Web sémantiques (SWS). D'une manière automatisée, les utilisateurs et les agents logiciels devraient pouvoir découvrir, appeler, composer, et surveiller l'exécution des services Web. Dans cet article, nous passons en revues les plus connues parmi elles à savoir OWL-S, SAWSDL et WSMO

## KEYWORDS

Services Web sémantiques (SWS), OWL-S, SAWSDL, WSMO

## 1. INTRODUCTION

Le Web est entrain de subir deux grandes évolutions à savoir les services Web et le Web sémantique. D'une part, l'évolution récente des technologies de l'Internet, guidée par le langage XML et ses technologies sous-jacentes, étend le rôle du Web d'un simple support d'information vers un intergiciel d'applications B2B. Cette nouvelle vague de l'Internet est guidée par le concept de services Web. Les services Web sont des applications auto-descriptives, modulaires et faiblement couplées qui fournissent un modèle simple de programmation et de déploiement d'applications, basé sur des normes, et s'exécutant au travers de l'infrastructure Web [KAD 03]. Les services Web fournissent des standards facilitent le transport, l'invocation, la description et la recherche. Le protocole SOAP (*Simple Object Access Protocol*) transporte les messages XML entre les services Web et permet également d'invoquer des services Web. Le langage WSDL (*Web Services Description Language*) offre la possibilité de décrire l'interface d'un service Web. Le standard UDDI (*Universal Description Discovery and Integration*) décrit l'annuaire dans lequel les services Web sont répertoriés d'une manière homogène. La recherche et la découverte automatique des services devraient être supportées par ce protocole. Néanmoins, le manque de sémantique dans WSDL empêche la découverte automatique et par conséquent l'invocation et la composition automatiques.

Le Web sémantique, quant à lui, est une extension du Web actuel, dans lequel les informations ont un sens bien défini afin de permettre aux ordinateurs et aux utilisateurs de travailler en coopération [BUR 01]. Le Web sémantique donne du sens au contenu des pages Web, contrairement au Web classique qui se contente de les visualiser. Le Web sémantique permet ainsi à un agent logiciel de comprendre ce contenu. Grâce à cette compréhension, l'agent est capable de parcourir des pages et réaliser des tâches telles que la recherche et l'extraction des informations en se basant sur les ontologies.

L'application du Web sémantique aux services Web a donnée naissance aux services Web sémantiques (SWS). D'une manière automatisée, les agents logiciels devraient pouvoir découvrir, appeler, composer, et surveiller l'exécution des services Web. Pour atteindre ce but, le SWS doit (1) offrir aux fournisseurs la possibilité de publier leurs services, (2) faciliter la tâche de l'annuaire afin qu'il puisse fournir et composer, de façon automatique, les services recherchés et (3) permettre au client d'invoquer et surveiller, de façon automatique, un service sélectionné par l'annuaire. Plusieurs approches ont été développées pour les SWS. Dans cet article, nous passons en revues les plus connues parmi elles à savoir OWL-S, SAWSDL et WSMO. Chacune de ces approches est motivée par un certain nombre d'objectifs. Néanmoins, elles ont en communs les deux objectifs suivant :

- *Découverte automatique de services Web* : La découverte automatique de services Web est un processus permettant de localiser, d'une manière automatique, des services Web qui fournissent des services répondants à un besoin spécifié par un client.
- *Invocation automatique des services Web* : En donnant seulement une description déclarative d'un service Web, un agent logiciel doit être capable d'invoquer automatiquement un service Web après l'avoir découvert.

Le reste de cet article est organisé de la façon suivante. Dans la Section 2 nous présentons l'approche OWL-S. La Section 3 est consacrée à l'approche SAWSDL et dans la Section 4

nous étudions l'approche WSMO. La Section 5 propose une synthèse autour de ces approches. Enfin, nous présentons quelques conclusions dans la Section 6

## 2. OWL-S

*Semantic Markup for Web Services* OWL-S [OWL 04] est un langage permettant de décrire les services Web de façon non ambiguë et interprétable par des programmes. Ce langage est basé sur le langage d'ontologie du Web (OWL). OWL-S est aussi une ontologie OWL particulière. Nous présentons dans la suite les motivations de OWL-S et son ontologie supérieure pour les services Web.

### 2.1. Motivations de OWL-S

En plus des motivations indiquées dans l'introduction, que les trois approches visent à réaliser, OWL-S s'applique à accomplir la *composition automatique des services Web*. L'accomplissement d'une tâche complexe implique la sélection, la composition automatique des services Web. Par exemple, l'utilisateur peut vouloir faire tout le nécessaire pour son voyage à une conférence. Il veut acheter un billet mais aussi il veut réserver un hôtel proche du lieu de la conférence. Il peut également ajouter des contraintes de coût minimal. Afin qu'un agent puisse exécuter ce type de tâche, OWL-S fournit une spécification des pré-requis et des conséquences de l'exécution de chaque service individuel ainsi qu'un langage pour décrire les services composés et le flux de données. Pour atteindre ces objectifs, OWL-S définit une ontologie supérieure pour la description, l'invocation et la composition des services Web. Celle-ci est présentée dans ce qui suit.

### 2.2 Ontologie supérieure de OWL-S

La structuration de l'ontologie supérieure de OWL-S est motivée par la nécessité de fournir trois types d'information essentiels pour un service, à savoir :

- *Que fait le service* : cette information est donnée dans le *Service Profile*. Ce dernier permet la description, la publication et la découverte des services, en spécifiant une description textuelle à destination des utilisateurs "humains", des propriétés fonctionnelles et des propriétés non fonctionnelles. Dans l'approche OWL-S, la section *Profile* est utilisée à la fois par les fournisseurs pour publier leurs services et par les clients pour spécifier leurs besoins. Par conséquent, elle constitue l'information utile pour la découverte et la composition de services [OWL 04]. Les recherches de services peuvent se baser sur n'importe quel élément de *Profile* comme critère [CHA 03];
- *Comment utilise-t-on le service* : cette information est fournie dans le *Service Model* qui est utilisé pour, entre autres, composer les services. OWL-S modélise les services en tant que processus et celui-ci est défini par ses entrées/sorties. Trois types de processus existent : les processus atomiques (*AtomicProcess*), simples (*SimpleProcess*) et composites (*CompositeProcess*). Un *AtomicProcess* représente le niveau le plus fin pour un processus et correspond à une action que le service peut effectuer en une seule interaction avec le client. Les *CompositeProcess* sont décomposables en d'autres processus (composés ou non) ; leur décomposition peut être spécifiée en utilisant un ensemble de structures de contrôles tels que : *Sequence*, *Split*, *If-Then-Else* etc. Un *SimpleProcess* est utilisé pour fournir une vue d'un processus atomique ou une représentation simplifiée d'un processus composite;

- *Comment accède-t-on au service* : cette information est donnée dans le *Service Grounding*. Celui-ci indique comment accéder concrètement au service et fournit les détails concernant les protocoles, les formats de messages et les adresses physiques...

### 3. SAWSDL

SAWSDL est un langage sémantique de description de service Web. Il est évolutif et compatible avec les standards des services Web existants, et plus spécifiquement avec WSDL. SAWSDL augmente l'expressivité du langage WSDL [CHR 01] avec la sémantique en utilisant des concepts analogues à ceux utilisés dans OWL-S [OWL 04]. SAWSDL fournit un mécanisme permettant d'annoter sémantiquement les types de données, les opérations, les entrées/sorties de WSDL. Nous présentons dans ce qui suit les motivations qui étaient à son origine.

#### 3.1. Motivations de SAWSDL

En plus de la découverte et l'invocation automatiques des services Web, déjà citées dans l'introduction, SAWSDL vise la réalisation des objectifs suivants :

- *Un langage au dessus des standards existants* : Les entreprises ont déjà investi dans des projets d'intégrations basés sur les services Web et utilisant les standards existant. Par conséquent, les concepteurs de SAWSDL considèrent que n'importe quelle approche pour la description sémantique des services Web devrait être compatible avec l'existant. A cet effet SAWSDL, une extension sémantique de WSDL a été conçue,
- *Concevoir un langage incrémental* : il est relativement facile de modifier les outils existants autour de WSLD afin qu'ils s'adaptent à SAWSDL. Ce qui fait de SAWSDL une approche incrémentale,
- *L'annotation doit être indépendante du langage de représentation de la sémantique* : Il y a un certain nombre de langages potentiels pour représenter la sémantique comme *Web Service Modeling Language* (WSML) [De 05] et *Unified Modeling Language* (UML) [OMG 07]. Chaque langage offre différents degrés d'expressivité. La position des concepteurs de SAWSDL est qu'il n'est pas nécessaire d'attacher les standards des services Web à un langage sémantique particulier. Cette approche est la vision décrite par Sivashanmugam et al. dans [SIV 03] et donne plus de flexibilité aux développeurs.

Dans ce qui suit nous présentons comment l'annotation est faite dans SAWSDL.

#### 3.2. Annotation sémantique

L'annotation sémantique des documents WSDL est possible grâce à l'extensibilité de WSDL 2.0. En effet, conceptuellement WSDL 2.0 est doté des constructions suivantes : *interface*, *opération*, *message*, *binding*, *service* et *endpoint*. Les trois premiers à savoir *interface*, *opération* et *message* concernent la définition abstraite du service tandis que les trois autres sont relatifs à l'implémentation du service. SAWSDL fournit des mécanismes pour référencer des concepts de modèles définis à l'extérieure du document WSDL. Cela se fait grâce à l'attribut «sawSDL». Il existe trois extensions de cet attribut. La première est *modelReference* et permet d'associer un composant WSDL ou XML Schema à un concept d'une ontologie. Les deux autres sont *liftingSchemaMapping* et *loweringSchemaMapping* et permettent de spécifier le mapping entre les données sémantiques et les éléments XML. Les *SchemaMapping* sont utilisés pour établir la correspondance entre les structures des

entrées et des sorties, et est utile lorsque les structures XML demandées par le client et celles fournies par le service sont différentes. L'annotation des *interfaces*, *opérations*, *entrées/sorties* et les *types XML simple* s'effectue en leurs associant un concept dans une ontologie par le biais de l'attribut `modelReference`. Cependant, l'annotation des types de données XML complexes peut nécessiter en plus un *Schema Mapping*. En effet, deux services Web peuvent manipuler le même type complexe mais avec deux structures différentes.

## 4. WSMO

WSMO est une ontologie qui décrit les différents aspects relatifs à la composition dynamique des services Web et se base sur WSMF [FEN 02] (*Web Service Modelling Framework*) qui spécifie les éléments principaux pour décrire les services Web sémantiques. De tels éléments incluent *ontologies*, *buts*, *services Web* et *médiateurs*. Les *ontologies* définissent la terminologie utilisée par les autres éléments. Les *buts* indiquent ce que l'utilisateur attend du service. Les *services Web* définissent les fonctionnalités offertes. Les *médiateurs* lient les différents éléments afin de permettre l'interopérabilité entre les composants hétérogènes. Le langage WSML [De 05] est utilisé pour décrire formellement tous les éléments de WSMO et l'environnement d'exécution WSMX [GRO 05] permet la découverte, la sélection, la médiation et l'invocation des services Web sémantiques.

### 4.1. Motivations de WSMO

WSMO partage avec OWL-S les mêmes motivations à savoir : la découverte, l'invocation et la composition automatiques des services Web. Cependant WSMO ajoute à celles-ci l'objectif suivant : *Un découplage fort entre les composants et un rôle central pour la médiation*. L'un des principes fondamentaux de WSMO consiste en la séparation totale entre les différents éléments impliqués dans la composition des services Web. A cet effet, WSMO veut distinguer entre comment le client formule sa demande et comment le fournisseur expose son service. Sachant que la demande et l'offre sont décrites de façons différentes, un travail important doit être effectué afin d'établir la correspondance entre la demande et l'offre. Ce travail est le rôle des médiateurs.

### 4.2. Les facettes de WSMO

Les facettes de WSMO sont les *ontologies*, les *médiateurs*, les *services Web* et les *buts*. Les ontologies fournissent la sémantique compréhensible par une machine pour les informations utilisées par tous les acteurs d'un service Web. WSMO permet d'importer une ontologie dans une autre ontologie soit directement, quand il n'y a pas de conflits, soit indirectement, par le biais d'un médiateur qui va résoudre les conflits possibles. Les blocs de base d'une ontologie sont *concepts*, *relations*, *fonctions*, *instances* et *axioms*. Un ensemble de propriétés non fonctionnelles est donnée généralement au début de chaque définition d'une ontologie. Les médiateurs permettent de lier différentes ressources hétérogènes et résoudre les incompatibilités à plusieurs niveaux. Il peut y avoir une incompatibilité entre les données ou entre les processus. Dans le premier cas, la médiation sert à établir la correspondance entre les différentes terminologies. Le deuxième type d'incompatibilité apparaît au moment de la communication entre différents services Web hétérogènes et dans ce cas, le médiateur fournit la fonctionnalité pour une analyse

d'exécution de deux services Web donnés, et compense les éventuelles disparités. Les services Web WSMO sont composés d'un ensemble optionnel de propriétés non fonctionnelles, un mécanisme permettant d'importer des ontologies soit directement en utilisant `importsOntology`, soit indirectement via un médiateur et une interface décrivant le comportement du service vis-à-vis de ses partenaires.

## 5. Synthèse

Avant de procéder à la comparaison des trois approches décrites ci-dessus, il convient de rappeler qu'aucune parmi celles-ci n'a atteint les objectifs initialement escomptés. En les évaluant par rapport aux objectifs communs, découverte et invocation automatiques, nous observons que ces deux objectifs sont partiellement pris en compte par OWL-S. D'une part, la découverte basée sur la sémantique est faite grâce à des outils, implémentant des Algorithmes de *Matching*, tel que *Matcher* [OWL 07] et d'autre part l'invocation automatique est réalisée en utilisant WSDL. WSMO et SAWSDL permettent uniquement l'invocation automatique. OWL-S étant le premier langage, les langages qui l'ont suivi tel que WSMO et SAWSDL se sont inspirés de lui, tout en évitant ses défauts. L'une des principales critiques faites à OWL-S est qu'il n'impose aucune contrainte afin que le *Service Profile* et le *Service Model* soient cohérents. Etant donné que OWL-S et WSMO sont proches l'un de l'autre, nous allons les comparer et ensuite nous comparons ces deux langages avec SAWSDL. Quoique WSMO introduise une nouvelle composante, les médiateurs, dans l'architecture des services, il reste proche de OWL-S. En effet, (1) le *Service Profile* de OWL-S ressemble aux deux composantes, *capacité* et *Propriétés non fonctionnelles* de WSMO, (2) le *Service Model* est similaire à une interface dans WSMO et (3) un *Objectif* WSMO étant composé d'une *interface*, une *capacité* et des *propriétés non fonctionnelles*, il ressemble donc à une partie du *Service Profile* et une partie *Service Model*. Néanmoins, en comparant le *Service Profile* et l'*Objectif* nous constatons que ce dernier est plus avantageux. En effet, l'*Objectif* est défini indépendamment du service et ainsi le même *Objectif* peut être utilisé par plusieurs services afin d'effectuer une recherche, tandis que le *Service Profile* est rattaché à un service donné. La dernière ressemblance est celle de *Service Grounding* de OWL-S et le *grounding* dans WSMO. Le médiateur de WSMO n'a pas de correspondant dans OWL-S.

S'agissant de SAWSDL, celui-ci s'intéresse à la découverte et l'invocation automatique des services mais il ne s'occupe pas de la composition. Etant proche de WSDL, SAWSDL ne nécessite pas beaucoup d'efforts pour les développeurs habitués à WSDL contrairement à OWL-S et WSMO. SAWSDL permet d'utiliser tous types d'ontologies (OWL, WSML [De 05] et UML [OMG 07]) tandis que OWL-S accepte des ontologies OWL et WSMO des ontologies WSML. Enfin, SAWSDL ne permet pas de définir des propriétés non fonctionnelles. En termes d'outillage, OWL-S tire partie de son ancienneté et ainsi il dispose de plusieurs outils allant du simple éditeur au composeur semi-automatique en passant par les outils de *Matching* et de validation. Cependant WSMO ne dispose que des outils pour l'édition, *WSML Editor* et un seul outil propre à SAWSDL existe, *SAWSDL Tool Annotation* (un outil pour l'annotation sémantique). Les outils pour WSMO s'avèrent plus difficiles à développés car celui-ci se base sur WSML, un langage qui n'a pas été utilisé auparavant, tandis que OWL-S et SAWSDL s'appuient sur RDF et XML. Le tableau suivant fournit un récapitulatif de cette comparaison.

	OWL-S	SAWSDL	WSMO
Type d'ontologie accepté	OWL	Tous	WSML
Composition des services	+/-	-	+/-
Outils	matcher, éditeurs	Outil d'annotation validateurs...	éditeurs
Médiation et buts	-	-	+

## 6. Conclusion

Dans cet article, nous avons exposé trois approches pour la description sémantique des services Web, à savoir : OWL-S, SAWSDL et WSMO. Ces approches constituent une bonne base pour le SWS malgré qu'il reste beaucoup de travail afin qu'elles soient utilisables. Toutes ces approches sont en soumission auprès du W3C [W3C 07]. Mais dernièrement le groupe de travail de SAWSDL a pratiquement atteint son objectif à propos de SAWSDL. Ce dernier tirera profit des mécanismes d'extension de WSDL 2.0 pour établir un soutien simple et générique consistant à ajouter des descriptions sémantiques pour les services de Web. Le W3C vient de finaliser WSDL 2.0, le 27 juin 2007 [W3C 07]. Il est le langage de description de services Web qui prend en compte le principal protocole Web, HTTP, ainsi que le protocole de services Web le plus couramment utilisé, SOAP. WSDL 2.0 intègre les correctifs apportés à WSDL 1.1. Les spécifications de SAWSDL développées par ce groupe de travail sont devenues une proposition pour une recommandation le 5 juillet 2007 [W3C 07].

## REFERENCES

- [BUR 01] BURNERS-LEE T., HENDLER J., LASSILA O., «The semantic Web», Scientific American, vol. 284, no 5, 2001.
- [CHA 03] CHARLET J., LAUBLET P., REYNAUD C., «Web sémantique », Rapport de recherche, octobre 2003, Action spécifique 32 CNRS / STIC, France.
- [CHR 01] CHRISTENSEN E., CURBERA F., MEREDITH G., WEERAWARANA S., Web Services Description Language (WSDL) Version 1.1 », Mars 2001, W3C Note.
- [DE 05] DE BRUIJN J. ET AL., «The Web Service Modeling Language», 2005, DERI.
- [DUB 07] Dublin Core Metadata Element Set, Version 1.1, The DCES Home Page,
- [FEN 02] FENSEL D., BUSSLER C., « The Web Service Modeling Framework WSMF », Electronic Commerce Research and Applications, vol. 1, no 2, 2002, p. 113–137.
- [GRO 05] WORKING GROUP W., « Overview and Scope of WSMX », 2005, DERI.
- [KAD 03] KADIMA H., MONFORT V., « Les Web Services », mars 2003, page 1.
- [OMG 07] OMG, « Unified Modeling Language (UML) Version 2.1.1 », [<http://www.omg.org/technology/documents/formal/uml.htm>], [référence du août 2007].
- [OWL 04] OWL-S COALITION, « Bringing Semantics to Web Services : The OWL-S Approach. », Semantic Web Services and Web Process Composition Workshop, vol. 3387 de Lecture Notes in Computer Science, San Diego, CA, USA, juillet, 2004, p. 26-42.
- [OWL 07] OWLSM, The TUB OWL-S Matcher, The Home Page,
- [SIV 03] SIVASHANMUGAM K., VERMA K., SHETH A., MILLER J., « Adding Semantics to Web Services Standards », ICWS'03, Nevada, juin, 2003, p. 395-401.
- [W3C 07] W3C, World Wide Web Consortium, The Website, [<http://www.w3c.org/>]