



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bericht INF-M3 AW2

Amine El Ayadi
Semantic Web Services from an Agent Perspective

Amine El Ayadi
Semantic Web Services from an Agent
Perspective

Ausarbeitung eingereicht im Rahmen der Veranstaltung Anwendung 2
im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Kai von Luck

Abgegeben am 28. Februar 2009

Amine El Ayadi

Thema

Semantic Webservices from an Agent Perspective

Stichworte

Semantic Web, Semantic Web Services, Agenten, Softwareagenten

Kurzzusammenfassung

Das derzeitige World Wide Web dient als Kommunikationsmittel für zwei Technologien: Semantic Web und Web Services. Web Services ist eine plattformunabhängige Technologie, die es ermöglicht, die Funktionalitäten eines Dienstes und ihre Schnittstellen über das Web zu veröffentlichen und zu nutzen. Da Semantic Web es ermöglicht, die Ressourcen des Internets für Maschinen verständlich zu machen, wird es zur Automatisierung der Nutzung von Webdiensten verwendet. Die Semantic Web Services stellen eine Zusammenführung von Semantic Web und Web Services dar, deren verschiedene Technologien in dieser Arbeit analysiert werden. Außerdem werden Dienste anhand der Web Service Modeling Ontologie (WSMO) semantisch beschrieben und ausgeführt. Das World Wide Web ist ein ideales Einsatzgebiet für Softwareagenten: (Menschliche) Anfragen zu interpretieren und genauer zu formulieren einerseits, Dienste zu finden, Inhalte von Dienstbeschreibungen zu erkennen gegen die Anfrage zu prüfen andererseits, anschließend die passende Dienste zu verwenden.

Amine El Ayadi

Title of the paper

Semantic Webservices from an Agent Perspective

Keywords

Semantic Web, Semantic Web Services, Agent, Softwareagent

Abstract

The current World Wide Web serves as a means of communication for two technologies: Semantic Web and Web Services. Web Services is a platform independent technology, which enables the publication and the use of the service functionality with their interfaces. Because Semantic Web enables the understanding of Internet resources by machines, it is used for the automation of Web Services processing and mainly the discovery of services. This combination of both technologies is Semantic Web Services, whose different technologies will be analysed in this work. Moreover, services are semantically described and executed on the basis of Web Service Modeling Ontology (WSMO). The World Wide Web is an ideal application area for software agents: to interpret und formulate (Human) requests more precisely, to find services, to recognize the content of service descriptions and to connect these Services.

Inhaltsverzeichnis

Inhaltsverzeichnis	4
1. Einführung und Motivation	5
1.1. Einführung	5
1.2. Ziele	6
1.3. Gliederung.....	6
2. Semantic Web	7
2.1 Resource Description Framework.....	8
2.2 DARPA Agent Markup Language.....	9
2.3 Ontology Inference Layer	9
2.4 Web Ontology Language.....	9
3. Semantic Web Services	10
3.1 Web Ontology Language-Services.....	11
3.2 Web Service Modeling Ontology.....	11
3.3 Semantic Web Services Framework.....	13
3.4 Web Services Description Language-Semantic	13
4. Analyse der Ansätze für Semantic Web Services	14
4.1 Ausdrucksmächtigkeit.....	14
4.2 Automatisierung.....	15
4.3 Kommunikation	15
4.4 Ansatz.....	16
4.5 Konzeptionelles Modell	16
4.6 Implementierung	17
5. Fazit	17
Quellen	18

1. Einführung und Motivation

1.1. Einführung

Das Internet hat sich in den letzten Jahren zu einem Massenmedium entwickelt und stellt eine riesige und weit verbreitete Informationsquelle dar. Hier werden Dokumente und andere Informationsobjekte von Benutzern veröffentlicht, die mittels Suchmaschinen (z.B. „Google“, „Yahoo“, „Msn“) von anderen Benutzern aufgefunden und aufgerufen werden können. Eine Suche mit maschinenverständlicher Bedeutung ist jedoch nicht möglich, da die im Web enthaltenen Informationen sich nicht maschinell verarbeiten lassen. Die derzeitigen Suchmaschinen sind nicht in der Lage, präzise zu suchen, da sie die Inhalte der Informationen nicht interpretieren können. Sie suchen durch Schlüsselwörter, und dabei werden alle möglichen Webseiten, auf die eines der Schlüsselwörter zutrifft, bzw. auf denen sie verzeichnet sind, dem Nutzer angezeigt. Dies ist für den Nutzer zeitaufwendig, da er die Angebote selbst vergleichen muss, um das beste mögliche Angebot zu finden.

Das Semantic Web und Semantic Web Services

Die Vision des Semantic Webs, die von Timothy Berners-Lee¹ (der Erfinder des World Wide Web) geprägt wurde, ist nun, das im Web enthaltene Wissen nicht nur für Menschen verständlich zu machen, sondern auch für Maschinen (Rechner). Damit soll eine automatische Nutzung von Information erreicht werden. Web Services ist neben dem Semantic Web eine weitere Technologie, die sich dem Internet als Kommunikationsmittel bedient. Hierbei handelt es sich um Dienste, die auf einem Server laufen. Derzeitige Web Services bieten Standards, die es ermöglichen, Dienstangebote manuell zu publizieren, zu suchen, zu finden und aufzurufen. Mit Hilfe von Semantic Web wird die Nutzung von Diensten in Web Services automatisiert, indem diese semantisch beschrieben werden. Dieser Ansatz wird „Semantic Web Services“ („semantisch beschriebene Webdienste“) genannt. Damit sollen Dienste nicht mehr manuell genutzt werden, sondern maschinell.

Semantic Web Services bieten eine Reihe von Vorteilen. Diese Vorteile resultieren einerseits aus dem Semantic Web und andererseits aus den Web Services. Im Web können die darin enthaltenen Informationen präsentiert, aber nicht maschinell verarbeitet werden. Dieser Mangel wird von Semantic Web kompensiert. Als Erweiterung des Internets stellt Semantic Web formale Sprachen zur Beschreibung der Daten bereit. Dies führt dazu, dass die Daten maschinell verarbeitet werden und die semantische Interoperabilität gewährleistet ist. Web Services ermöglichen durch ihre verwendeten Standards eine syntaktische Beschreibung der Daten (syntaktische Interoperabilität). Als weitere Vorteile werden vor allem die Unabhängigkeit vom verwendeten Betriebssystem und der verwendeten Programmiersprache genannt. Durch die Integration von Semantic Web in Web Services wird die Suche nach einem Dienst im Dienstverzeichnis verbessert, da das Dienstverzeichnis semantische Informationen enthält und der Dienstanutzer durch *Softwareagenten* ersetzt wird [IEEE 01].

Softwareagenten

Ein Softwareagent ist ein Stück Software, das im Wesentlichen folgende Eigenschaften hat: Nach Übermittlung der Aufgabe handelt der Agent *autonom* und *proaktiv*, zur Lösung der Aufgabe *kommuniziert* er mit anderen Agenten und ggf. handelt er *reaktiv* in seiner Umwelt, er ist eventuell *mobil*, kann sich veränderten Umfeldern und Fragestellungen *anpassen*, *lernt* im Laufe seines Lebenszyklus dazu, insgesamt wirkt sein Handeln auf einen Beobachter *intelligent* [IEEE 01].

„Software-Agenten sind die Benutzerschnittstelle zum Semantic Web. Als virtuelle Handlungsreisende bevölkern sie das Semantic Web und führen für ihre menschlichen Benutzer Aufträge aus. Trotz der zentralen Bedeutung von **Softwareagenten** unterhält das World Wide Web Consortium (W3C) keine separate Aktivität in diesem Bereich. Dies mag auf den ersten Blick erstaunen. Bei genauerem Hinsehen wird aber deutlich, dass sich die Entwickler des Semantic Web an Vorarbeiten an Software-Agenten orientieren und das W3C mit den für das Semantic Web spezifischen Empfehlungen (Recommendations) gleichsam Bedingungen für Software-Agenten formuliert" [GRT 2006]

1.2. Ziele

Durch das großes Interesse am Bereich von Semantic Web Services gibt es verschiedene Ansätze, an denen derzeit geforscht wird:

- Web Services Description Language-Semantic (WSDL-S),
- Semantic Web Services Framework (SWSF),
- Web Ontology Language-Services (OWL-S) und
- Web Service Modeling Ontology (WSMO).

Im Moment ist noch nicht abzusehen, welcher dieser Ansätze sich in der Zukunft durchsetzen wird, eine Analyse ist jedoch möglich. Die Analyse soll die Technologien nach bestimmten Kriterien bewerten und zum Schluss den Ansatz, der die Anforderungen am besten erfüllt, auswählen. Das Ziel dieser Arbeit liegt daher darin, diese Analyse durchzuführen. Das weitere Ziel dieser Arbeit besteht darin, fest zu stellen, inwieweit ein Dienstinutzer durch **Softwareagenten** ersetzt werden kann [ACM 01].

1.3. Gliederung

Kapitel 1 leitet die Arbeit mit der Erläuterung der Motivation für diese Arbeit ein. Außerdem werden die Ziele dieser Arbeit vorgestellt. Im zweiten Kapitel werden die Konzepte und Technologien vom Semantic Web vorgestellt. In Kapitel 3 werden die Lösungsansätze für Semantic Web Services vorgestellt, analysiert und verglichen, dazu wird der Einsatz von **Softwareagenten** als Dienstinutzer untersucht.

Kapitel 4: Zusammenfassung und Ausblick. Hier wird die Arbeit zusammengefasst und ein Ausblick auf mögliche Erweiterungen oder Verbesserungen gegeben.

2. Semantic Web

“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.” [BHL01].

In diesem Zitat von Berners-Lee, Hendler und Lassila, das 2001 veröffentlicht wurde, wird das Konzept des Semantic Webs in wenigen Worten zusammengefasst. Dabei soll das Semantic Web ermöglichen, dass die sich im Web befindende Datenmenge auch für Maschinen (Rechner) interpretierbar wird.

Der Mensch hat die kognitive Fähigkeit, Informationen, die auf Webseiten veröffentlicht sind, zu interpretieren. Der Computer dagegen hat diese Fähigkeit nicht. Die Idee des Semantic Webs ist nun, diese Informationen so zu formulieren, dass sie vom Computer eindeutig interpretiert und weiterverarbeitet werden können. Das Semantic Web bedient sich hierzu der grundlegenden Repräsentationssprache des Internets. Die Informationen werden dabei in Form von Ontologien erfasst. Wie sich diese Ontologie in das Schichtenmodell des Semantic Webs integriert, wird in der Abbildung 2.1 gezeigt.

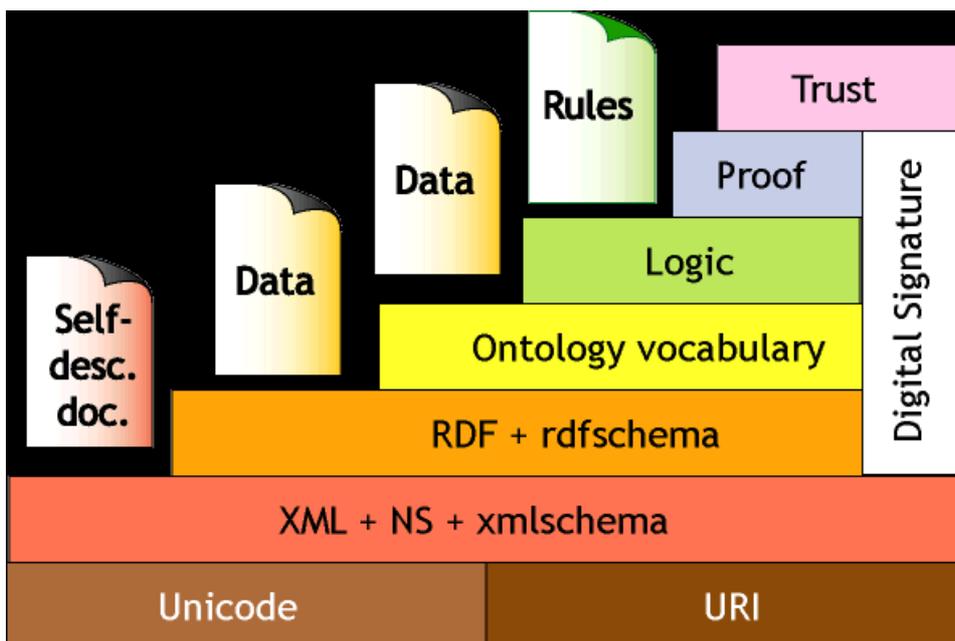


Abbildung 2.1: Semantic-Web-Schichtenmodell aus [BCH+01]

Im Folgenden werden die verschiedenen Techniken, die in der Abbildung dargestellt sind, beschrieben:

Die *Unicode und URI-Schicht*. Der Unicode ist ein Standard für Computerdarstellung von Zeichen und der URI (Uniform Resource Identifier) dient dazu, Ressourcen im Web eindeutig zu identifizieren.

Die *XML-Schicht* ermöglicht erwähnt eine strukturierte Darstellung von Dokumenten.

Die *RDF und RDF-S Schicht* (siehe Abschnitt 2.1).

Die *Ontologieschicht* ermöglicht die Definition und die Darstellung der Beziehungen zwischen Ontologiekonzepten.

Die *Logik-Schicht* kontrolliert, dass die Daten nur in maschinenverarbeitbarer Form abgelegt werden können.

Die *Proof-Schicht* soll durch eine logische Beweisführung die Konsistenz der Daten sicherstellen.

Die *Trust-Schicht* soll die Richtigkeit der Daten überprüfen.

Die *Digital Signature* wird zur Authentifizierung des Absenders der Nachricht oder des Dokuments verwendet.

Die Ontologiesprachen des Semantic Webs werden in den Abschnitten 2.1, 2.2, 2.3 und 2.4 vorgestellt.

2.1 Resource Description Framework

Das Resource Description Framework (RDF) ist eine Sprache, die vom W3C entwickelt wurde, um Metadaten zur Beschreibung von Web-Ressourcen zu erzeugen [LaS99].

Das RDF-Modell ist einfach gehalten und besteht aus drei Objekttypen: Ressourcen, Eigenschaften und Aussagen. Eine Aussage besteht aus einem Tripel (Subjekt, Prädikat, Objekt).

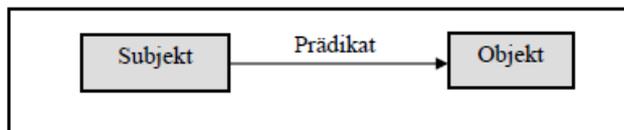


Abbildung 2.2: RDF-Modell

Das Subjekt ist die zu beschreibende Ressource, das Prädikat ist eine Eigenschaft dieser Ressource und das Objekt dient als ein Wert der Eigenschaft [SGA07]. Hier folgt ein Beispiel:

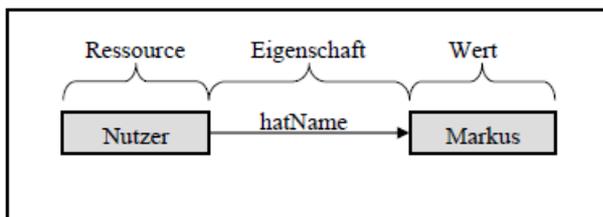


Abbildung 2.3: RDF-Beispiel

Das RDF-Schema (RDF-S) ist eine Erweiterung von RDF und definiert auf Basis von RDF spezielle Typen von Ressourcen und Eigenschaftselementen. RDF-S stellt Basisstrukturen für Klassen und deren Eigenschaften zur Verfügung.

2.2 DARPA Agent Markup Language

Die DARPA Agent Markup Language (DAML) wurde im Rahmen eines Forschungsprogramms der Defense Advanced Research Projects Agency in Jahr 2000 entwickelt. DAML ist eine formale Sprache, die benutzt wird, um die Integration der Ontologie im Semantic Web zu erleichtern.

2.3 Ontology Inference Layer

Ontology Inference Layer (OIL) ist als eine Erweiterung von RDF kreiert worden, mit dem Ziel, webbaserendes Wissen effizient und verständlich aufzubereiten [ACM02]. Weiterführende Informationen finden sich in [HFB+00]. OIL wird jetzt durch DAML+OIL ersetzt.

2.4 Web Ontology Language

Wie RDF (2.1.1) wurde die Web Ontology Language (OWL) vom W3C entwickelt. OWL ist eine Standardsprache zur Repräsentation von Informationen im Semantic Web. Sie stammt von DAML+OIL und besteht aus drei Sprachebenen (OWL Lite, OWL DL, OWL Full) mit ansteigender Ausdrucksmächtigkeit.

1. *OWL Lite* ist die Version mit der geringsten Ausdrucksmächtigkeit.
2. *OWL DL* besitzt maximale Ausdrucksmächtigkeit, ist eine Subsprache von OWL Full, deren Sprachkonstrukte teilweise im Gebrauch eingeschränkt sind.
3. *OWL Full* ist eine Obermenge von RDF und hat maximale Ausdrucksmächtigkeit. Sie besteht aus denselben Sprachkonzepten wie OWL DL, aber ohne Einschränkung.

3. Semantic Web Services

Die Beschreibung eines Dienstes (syntaktisch) kann von einem Menschen erkannt werden, aber eine Maschinenverarbeitbarkeit der Daten ist nicht möglich. Semantic Web Services löst dieses Problem durch die semantische Beschreibung der Daten.

Die Vision von Semantic Web Services (SWS) kann als eine Kombination von Web Services und den beschriebenen Technologien des Semantic Webs (Kapitel 2) beschrieben werden (Abbildung 3.1).

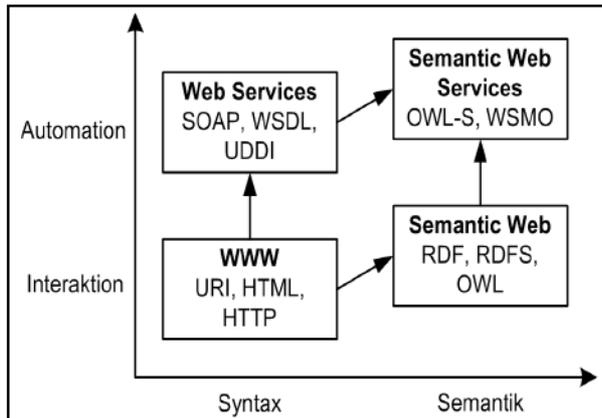


Abbildung 3.1: Semantic Web Services nach [DoJ04]

SWS als eine Erweiterung von Web Services ermöglicht das automatische Auffinden und Ausführen von Web Services, indem die semantische Annotation in der Dienstbeschreibung verwendet wird. Der Dienst ist semantisch so beschrieben, dass ein *Softwareagent* in der Lage ist, Dienstinformationen zu interpretieren und zu nutzen [MSZ01]. Es existieren vier Lösungsansätze: WSMO, OWL-S, WSDL-S und SWSF, die es ermöglichen, die Vision von SWS zu realisieren. Diese Ansätze werden zunächst vorgestellt und analysiert.

Lösungsansätze

Semantic Web Services stellen ein noch sehr dynamisches Forschungsfeld dar, dessen aktueller Stand in diesem Kapitel untersucht werden soll. Die Abbildung 3.1 zeigt die verschiedenen Lösungsansätze von SWS und die jeweilige Ontologiesprache, die ihren Einsatz ermöglichen. Diese Ansätze werden zunächst vorgestellt und beschrieben, anschließend werden sie im Abschnitt 4 bewertet.

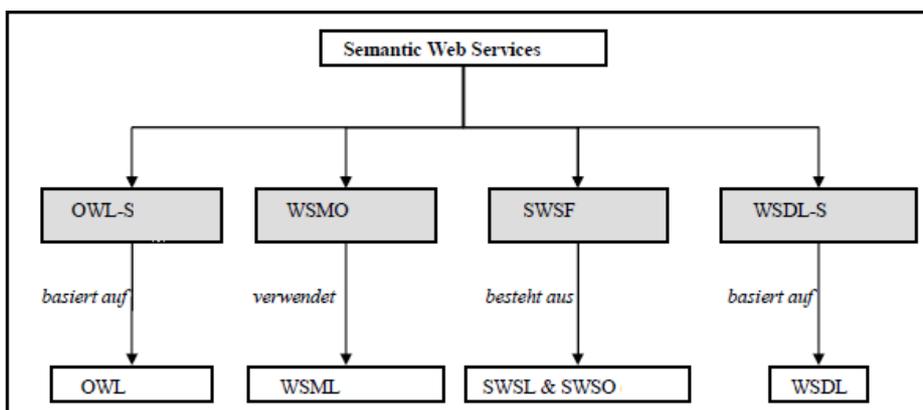


Abbildung 3.1: Organisation der Technologien

Um einen besseren Überblick über die verschiedenen Technologien zu ermöglichen, werden OWL-S, WSMO, SWSF und WSDL-S jeweils in den Abschnitten 3.1, 3.2, 3.3 und 3.4 beschrieben.

3.1 Web Ontology Language-Services

Seit der ersten Veröffentlichung des Vorgängers im Mai 2001 unter dem Namen DAML-S bis zur aktuellen Version 1.222, welche im März 2006 veröffentlicht wurde, hat die Web Ontology Language-Services (OWL-S) zahlreiche Verbesserungen und Weiterentwicklungen erfahren. Sie basiert auf OWL (2.2.4). OWL-S [MBH+04] besteht aus drei grundlegenden Elementen, die jeweils eine Teilontologie modellieren (siehe Abbildung 3.2):

1. Das *Service Profile* definiert, was der Dienst leistet. Dies gibt den *Softwareagenten* die Möglichkeit den Service zu finden und auszuwählen.
2. Das *Service Model* definiert das Ausführungsmodell eines Dienstes. Es legt also fest, wie ein Dienst arbeitet.
3. Das *Service Grounding* beschreibt, wie der Dienst verwendet wird, d.h. wie ein Kunde den Dienst aufrufen kann.

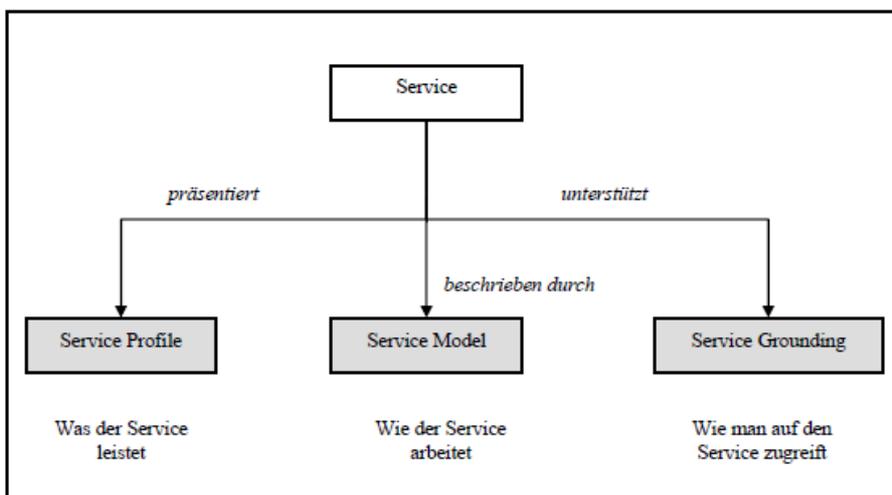


Abbildung 3.2: OWL-S Ontologien nach [MBH+04, Abb. 1]

3.2 Web Service Modeling Ontology

Um Web Service Modeling Ontology (WSMO) zu beschreiben, wird zunächst eine kurze Einführung gegeben. Anschließend werden seine Hauptelemente dargestellt. Diese Hauptelemente werden anhand der WSML-Sprache modelliert.

Die WSMO wurde im April 2005 als Standardvorschlag beim W3C eingereicht. Sie ist ein konzeptionelles Modell für semantische Servicebeschreibung und zugleich eine vollständige Beschreibungssprache zur Annotierung von Web Services [BBD+05]. Sie basiert auf WSMF und wurde davon abgeleitet. Ziel von WSMO ist es, Verfahren bereitzustellen, mit denen die semantische Dienstnutzung automatisiert werden kann.

Das WSMO-Projekt ist nach [CaS06] in drei Arbeitsgruppen unterteilt:

1. Die *WSMO Working Group* bietet ein konzeptionelles Modell zur Beschreibung von Diensten.
2. Die *WSML Working Group* definiert eine konkrete und formale Sprache für die Beschreibung von SWS in WSMO.
3. Die *WSMX Working Group* definiert und bietet die Ausführungsumgebung für SWS.

WSMO-Hauptelemente

WSMO basiert auf vier Hauptelementen: Ontologies, Goals, Web Services und Mediators, wie in Abbildung 3.5 dargestellt:

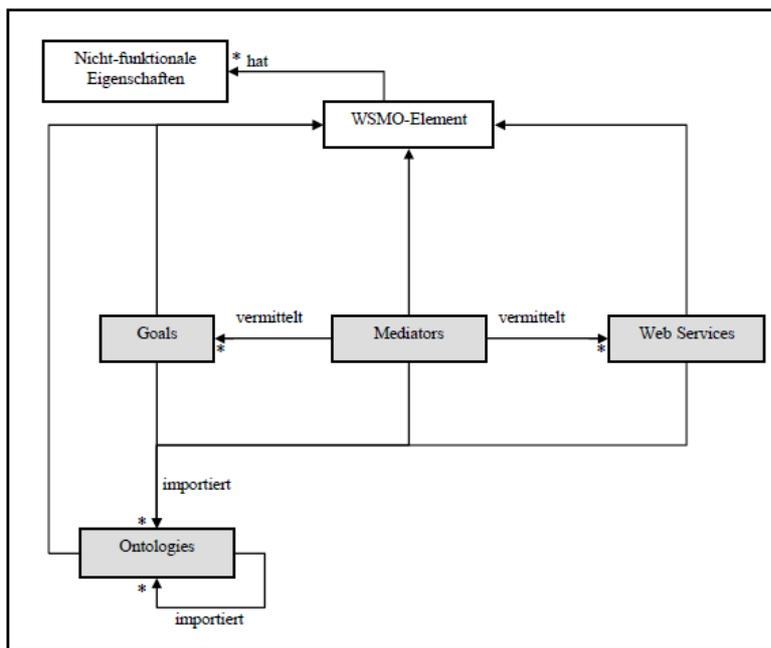


Abbildung 3.5: Die vier Hauptelemente des konzeptionellen Modells von WSMO nach [RLK05].

Anzumerken ist in diesem Zusammenhang ist, dass alle Hauptelemente (Ontologies, Web Services, Goals und Mediators) mehr oder weniger über nicht-funktionale Eigenschaften, importierte Ontologies und verwendete Mediators verfügen.

Ontologies

Die *Ontologies* bieten eine formale spezifizierte Terminologie, die von allen anderen Komponenten verwendet wird [StH05]. Die wesentlichen Elemente von WSMO-Ontologies sind *Konzept*, *Attribut*, *Instanz*, *Relation*, *Funktion* und *Axiom*.

Web Services

Unter Web Services in WSMO ist die semantische Beschreibung von angebotenen Diensten zu verstehen. Diese Beschreibung besteht aus einer Capability und einem Interface

Goals

Goals sind die gewünschten Funktionalitäten eines Dienstes, welche ein Dienstanutzer von einem Dienstanbieter erwartet. Sie werden genauso definiert wie WSMO-Web-Services d.h. durch eine Fähigkeit und ein Interface.

Mediators

Mediators haben zum Ziel, die Heterogenitäten zwischen den verschiedenen Hauptelementen von WSMO zu beseitigen. Dazu definiert WSMO verschiedene Typen von Mediators: OO-, GG-, WW- und WG-Mediators [BBD+05]. Dabei werden die WSMO-Hauptelemente durch diese Mediators verbunden.

3.3 Semantic Web Services Framework

Das Semantic Web Services Framework (SWSF) ist ein weiterer alternativer Standard im Bereich von SWS, welcher von der Semantic Web Services Initiative (SWSI) vorangetrieben wird. Es befindet sich seit Mitte 2005 in der Version 1.0 auf dem Markt und besteht (ähnlich wie WSML und WSMO) aus zwei Teilen: der Semantic Web Services Language (SWSL) und der darauf aufbauenden Semantic Web Services Ontology (SWSO).

Semantic Web Services Language

Die Semantic Web Services Language (SWSL) ist eine Logiksprache und besteht aus zwei Teilen: der SWSL-FOL (basierend auf Prädikatenlogik) sowie den SWSL-Rules (eine regelbasierte Sprache).

Semantic Web Services Ontology

Die Semantic Web Services Ontology (SWSO) ist die Ontologie für SWSL, mit deren Hilfe Services beschrieben werden können. Diese Ontologie hat zwei Teile: die First-order Logic Ontology for Web Services (FLOWS) und die Rules Ontology for Web Services (ROWS). SWSO hat einige Ähnlichkeiten zu OWL-S bei der Dienstbeschreibung, die ebenfalls dreigeteilt ist (Service Descriptors, Process Model und Grounding).

3.4 Web Services Description Language-Semantic

OWL-S, WSMO und SWSF definieren eine Sprache, um Webdienste zu beschreiben. Die Web Services Description Language-Semantic (WSDL-S) geht einen anderen Weg. Sie ist eine Erweiterung von WSDL, die es erlaubt, die Schnittstellenbeschreibung mit semantischen Konzepten zu annotieren, indem die Elemente in WSDL durch Verweise auf eine externe Ontologie mit semantischen Informationen verknüpft werden. Im November 2005 wurde WSDL-S zur Standardisierung beim W3C eingereicht.

Die Erweiterung von WSDL betrifft konkret die Eingaben, Ausgaben und Operationen eines Webdienstes, die auf einem Domänenmodell abgebildet werden. Die Operationen bestehen aus jeweils einer *Vorbedingung* und beliebig vielen Effekt-Elemente. Als Mechanismus für die Umwandlung von Informationen aus Ontologien und deren Äquivalenz in WSDL werden Transformationssprachen wie XSLT verwendet. Die Einfachheit und die große Akzeptanz der Sprache WSDL sind die wichtigsten Vorteile dieses Ansatzes gegenüber den bereits vorgestellten Ansätzen für SWS.

4. Analyse der Ansätze für Semantic Web Services

Im letzten Abschnitt wurden vier Ansätze zur semantischen Dienstbeschreibung erläutert: OWL-S, WSMO, SWSF und WSDL-S. Obwohl diese vier Technologien dasselbe Ziel haben, nämlich Webdienste semantisch zu beschreiben, weisen sie einige Unterschiede auf. In diesem Abschnitt geht es darum, diese Ansätze miteinander zu vergleichen. In den folgenden Tabellen soll eine Bewertung dieser vier vorgestellten Technologien erfolgen. Dabei bezeichnen die Zahlen 1 bis 4 die Erfüllungsgrade der Anforderungen (wobei 1 den höchsten und 4 den niedrigsten Grad darstellt) und die Kennzeichnung mit „-“ bedeutet, dass in diesem Fall die Anforderung nicht erfüllt wurde.

4.1 Ausdrucksmächtigkeit

Wenn eine Sprache ausdrucksstark genug ist, können existierende Dienste präzise beschrieben werden.

	WSDL-S	OWL-S	WSMO	SWSF
Sprache	-	OWL	WSML	SWSL
Ausdrucksmächtigkeit	4	3	2	1

Tabelle 4.1: Bewertung der Ausdruckskraft.

Im Gegensatz zu WSDL-S, die zur Beschreibung ihrer Dienste keine Ontologiesprache definiert, bauen die anderen Technologien jeweils auf einer bestimmten Sprache auf.

Die Abbildung 4.1 zeigt die Klassifizierung dieser Sprachen.

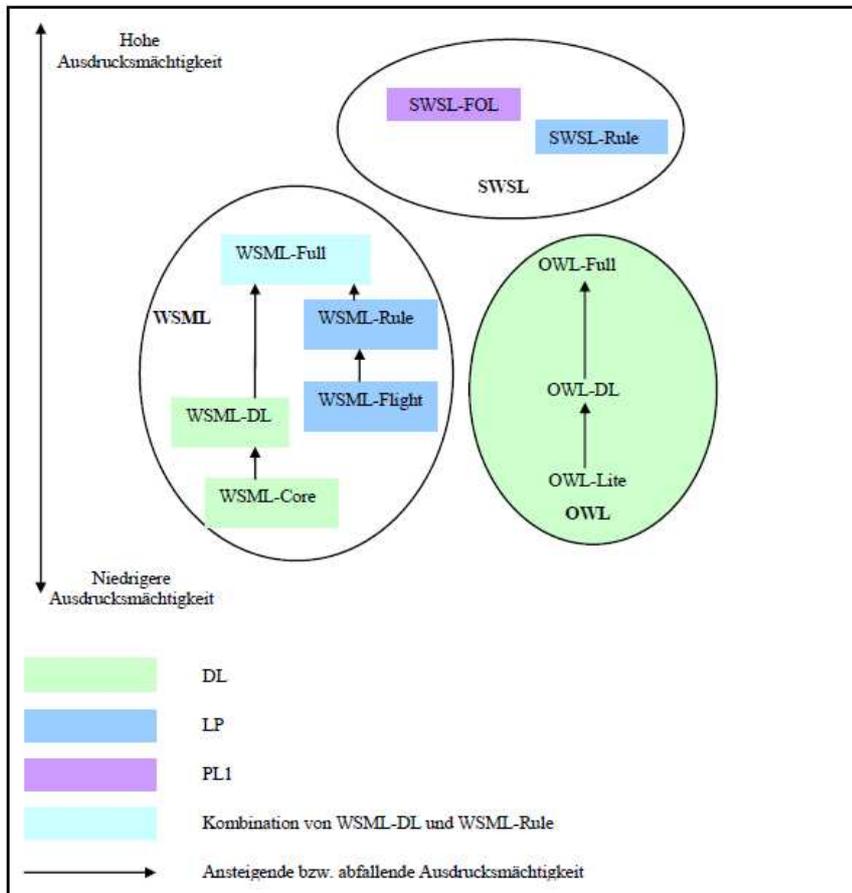


Abbildung 4.1: Klassifizierung von OWL, WSML und SWSL.

Dienstbeschreibung Zur Beschreibung der Dienste definieren alle Ansätze funktionale und nicht-funktionale Eigenschaften, außer WSDL-S, die nur eine nicht-funktionale Eigenschaft definiert. Die nicht-funktionalen Eigenschaften ermöglichen in SWS die Verfeinerung der Suche nach einem Service. In OWL-S und SWSF wird die Definition dieser nicht-funktionalen Eigenschaften in Service Profile bzw. in Service Description eingeschränkt. WSMO jedoch definiert sie in fast allen seiner Hauptelemente (Ontologies, Goals und Web Services). Dies ist ein wichtiger Vorteil für WSMO, da auf diese Weise die Suche nach einem Service erleichtert wird. Für WSDL-S wird die Suche somit erschwert.

	WSDL-S	OWL-S	WSMO	SWSF
Funktionale Eigenschaft	1	1	1	1
Nicht-funktionale Eigenschaft	3	2	1	2

Tabelle 3.3: Bewertung der Dienstbeschreibung.

4.2 Automatisierung

Es soll nicht mehr manuell nach Diensten im Web gesucht werden, wie es bei Web Services der Fall ist. Das automatische Suchen und Aufrufen sind die Hauptziele von SWS. Bei der automatischen Service Discovery geht es darum, einen Service mit Hilfe von *Agenten* im Dienstverzeichnis zu suchen. Dies wird bei WSMO, OWL-S und SWSF vollständig erfüllt, aber bei WSDL-S nicht.

Die automatische Service Invocation wird durch den Grounding-Mechanismus ermöglicht. Bei WSMO erfolgt dieser Mechanismus anhand des WSMO-Grounding und bei OWL-S mittels Service Grounding (siehe 3.1). Das Grounding von OWL-S ist ausgereifter als das von WSMO [ABB+04]. Da WSDL-S eine Erweiterung von WSDL ist, geschieht der Grounding-Mechanismus auf direktem Weg. Die Invocation bei SWSF erfolgt nur durch die Ein- und Ausgaben von atomaren Prozessen [BBB+05] und seine Ausführung ist noch nicht definiert, deswegen ist es hier mit „-“ kategorisiert.

	WSDL-S	OWL-S	WSMO	SWSF
Service Discovery	3	1	1	1
Service Invocation	1	2	3	-

Tabelle 3.4: Bewertung der Automatisierung.

4.3 Kommunikation

In offenen Systemen wie dem Internet herrscht sehr oft das Problem der Systemheterogenität, da verschiedene Systeme interagieren. Damit wird die Kommunikation in solchen Systemen behindert. WSMO ist der Einzige von den in dieser Arbeit vorgestellten Ansätzen, der dieses Problem explizit löst. Dazu werden Mediators verwendet. OWL-S und WSDL-S behandelt dieses Problem implizit durch Mapping-Verfahren, aber SWSF nicht.

WSMO und SWSF sind die Einzigen, die den Dienstanbieter und den Dienstanwender unterstützen, da sie die angebotenen Dienste und die Erwartungen des Nutzers getrennt definieren. Bei WSMO ist diese Trennung bei dessen Hauptelementen deutlich zu sehen, aber bei SWSF ist sie implizit [BBB+05]. Diese Trennung hat den Vorteil, dass es keine Abhängigkeit zwischen dem Dienstanwender und dem -anbieter gibt. Jedoch werden sie bei OWL-S und WSDL-S nicht getrennt.

	WSDL-S	OWL-S	WSMO	SWSF
Lösung der Heterogenitätsprobleme	2	2	1	-
Unterstützung der Dienstanbieter	2	2	1	1
Unterstützung der Dienstanutzer	2	2	1	1

Tabelle 3.5: Bewertung der Kommunikation.

4.4 Ansatz

OWL-S, WSMO und SWSF weisen einen Top-Down-Ansatz zur semantischen Dienstbeschreibung auf, WSDL-S hingegen einen Bottom-Up-Ansatz. Der Top-Down-Ansatz definiert die Web Services und ihre Semantik, unabhängig von den bestehenden WS-Technologien. Dies wird mittels einer Ontologiesprache ermöglicht. Der Bottom-Up-Ansatz dagegen versucht die bestehenden WS-Technologien um Semantik anzureichern. Die Vorteile des Bottom-Up-Ansatzes gegenüber dem Top-Down-Ansatz sind seine Einfachheit und seine Flexibilität. Das hat zur Folge, dass WSDL-S der einzige Ansatz ist, der WS-Standards voll und ganz akzeptiert. Somit wird die Interoperabilität zu bestehenden Systemen gewährleistet. Das SOAP-Protokoll, WSDL (wegen seiner Unterstützung bei der Service Invocation) und UDDI werden bei allen Ansätzen verwendet, außer WSMO und SWSF, die kein UDDI-Verzeichnis benutzt.

	WSDL-S	OWL-S	WSMO	SWSF
Ansatz	Bottom-Up	Top-Down	Top-Down	Top-Down
SOAP	1	1	1	1
WSDL	1	2	3	3
UDDI	1	1	-	-

Tabelle 3.6: Vergleichstabelle der verwendeten Ansätze.

4.5 Konzeptionelles Modell

Wegen des Bottom-Up-Ansatzes wird in WSDL-S nicht versucht ein neues konzeptionelles Modell zu erstellen wie etwa in OWL-S, SWSF und WSMO. Jedoch fehlt in OWL-S und SWSF ein einheitliches konzeptionelles Modell (es gibt mehrere unterschiedliche Definitionen für den zentralen Begriff „Service“). WSMO hingegen definiert sein Modell auf Meta Object Facility (MOF) als eine aus vier Schichten bestehende Metadaten-Architektur [FLP+07]:

1. Die Informationsschicht enthält die Daten, die beschrieben werden sollen.
2. Die Modellschicht enthält die Metadaten, die die Daten der Informationsschicht beschreiben (Ontologies, Goals, Web Services und Mediators).
3. Die Meta-Modellschicht beschreibt die Struktur und die Semantik der Metadaten (WSMO-Eigenschaften).
4. Die *Meta-Meta-Modellschicht* beinhaltet die eigentlichen Sprachdefinitionen von WSMO.

4.6 Implementierung

Bei der Implementierung kommt WSDL-S an die erste Stelle, da seine Entwicklungsumgebung (METEOR-S) einfacher ist zu benutzen als die von OWL-S (z.B.: OWL-S IDE) und WSMO (WSMX). Seine Einfachheit resultiert daraus, dass hier keine Mediatoren verwendet werden, wie es bei WSMX der Fall ist. Auch OWL-S benutzt explizit keine Mediatoren, belegt aber nur den dritten Platz, da hier viele Komponenten (OWL-S Matchmaker, JUDDI, OWL-S IDE, etc.) zusammenarbeiten müssen, was die Implementierung erschwert. In der Literatur wurde kein Tool für SWSF gefunden [BBB+05].

	WSDL-S	OWL-S	WSMO	SWSF
Entwicklungsumgebung	1	3	2	-

Tabelle 3.8: Bewertung der verwendeten Werkzeuge.

5. Fazit

	WSDL-S	OWL-S	WSMO	SWSF
Ausdrucksmächtigkeit	4	3	2	1
Dienstbeschreibung	3	2	1	2
Automatisierung	2	1	2	3
Kommunikation	3	3	1	2
Ansatz	1	2	4	4
Konzeptionelles Modell	3	2	1	2
Implementierung	1	3	2	4
Endbewertung	17	16	13	18

Tabelle 5.1: Endbewertung

Nach dieser Endbewertungstabelle zeigt SWSF die schlechtesten Ergebnisse. Außerdem existieren bislang keinerlei Implementierungen, die darauf basieren. Aufgrund seiner Einfachheit und der hohen Akzeptanz der bestehenden WS-Technologien hat WSDL-S einen Vorteil gegenüber OWL-S und WSMO. Aber WSDL-S besitzt zur Beschreibung seiner Dienste nur eine nicht-funktionale Eigenschaft. Da für eine semantische Dienstsuche die nicht-funktionalen Eigenschaften wichtig sind, sind in dieser Hinsicht OWL-S und WSMO als besser zu bewerten.

Nach der oberen ist WSMO die beste Technologie für SWS. Zusätzlich ist WSMO konzeptionell gesehen stärker als OWL-S.

Quellen

- **[IEEE 01] Web Services from an Agent Perspective**, 2008
Terry R. Payne (University of Southampton)
[\[http://ieeexplore.ieee.org/Xplore?url=/iel5/9670/4475846/04475853.pdf\]](http://ieeexplore.ieee.org/Xplore?url=/iel5/9670/4475846/04475853.pdf)
- **[ACM 01] Virtual web services: application of software agents to personalization of web services.** *Jarogniew Rykowski, Wojciech Cellary* (The Poznan University of Economics, Poland)
[\[http://portal.acm.org/ft_gateway.cfm?id=1052272&type=pdf&coll=Portal&dl=ACM&CFID=23776836&CFTOKEN=57582297\]](http://portal.acm.org/ft_gateway.cfm?id=1052272&type=pdf&coll=Portal&dl=ACM&CFID=23776836&CFTOKEN=57582297)
- **[ACM 02] Ontology support for web service processes**
C. Pahl, M. Casey (Dublin City University, Dublin, Ireland)
[\[http://portal.acm.org/ft_gateway.cfm?id=940099&type=pdf&coll=Portal&dl=ACM&CFID=23776836&CFTOKEN=57582297\]](http://portal.acm.org/ft_gateway.cfm?id=940099&type=pdf&coll=Portal&dl=ACM&CFID=23776836&CFTOKEN=57582297)
- **[GRT 01] Software-Agenten im Semantic Web.** Springer Informatik Spektrum
29/1 Feb 2006 *Grütter R*
[\[http://www.springerlink.com/content/71513482h4551q13/?p=f35c8ee8ea5047bc8f8993890ca4298cπ=2\]](http://www.springerlink.com/content/71513482h4551q13/?p=f35c8ee8ea5047bc8f8993890ca4298cπ=2)
- **[BHL01]The Semantic Web.** Scientific American, 2001.
T. Berners-Lee, James Hendler und Ora Lasilla
- **[BCH+01]. W3C Semantic Web Activity.**
T. Berners-Lee, D. Connolly, S. Hawke, I. Herman, E. Prud'Hommeaux und R. Swick
[\[http://www.w3.org/2001/sw, Jul. 2008\].](http://www.w3.org/2001/sw, Jul. 2008)
- **[LaS99] Resource Description Framework (RDF) Model and Syntax Specification.** W3C Recommendation, 1999.
O. Lassila und R. R. Swick.
[\[http://www.w3.org/TR/1999/REC-rdf-syntax-19990222\]](http://www.w3.org/TR/1999/REC-rdf-syntax-19990222), Mär. 2008.
- **[SGA07] Semantic Web Services: Concepts, Technologies and Applications.** Springer, 2007.
R. Studer, S. Grimm, A. Abecker.
- **[HFB+00] The Ontology Inference Layer: OIL. Technical Report IR-479.**
Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, R. Studer und E. Motta (Vrije University of Amsterdam, Faculty of Sciences, 2000)
- **[DoJ04] Semantik, Odem einer Service-orientierten Architektur.** In Java Spektrum, 2004.
W. Dostal und M. Jeckle.
[\[http://www.jeckle.de/semanticWebServices/intro.html\]](http://www.jeckle.de/semanticWebServices/intro.html), Jul. 2008.
- **[MSZ01] Semantic Web Services. IEEE Intelligent Systems, Special Issue on the Semantic Web**, 16(2):46-53, 2001. *S. A. McIlraith, T. Cao Son, und H. Zeng.*

- **[MBH+04] OWL-S: Semantic Markup for Web services. 2004.**
D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan und K. Sycara.
<http://www.daml.org/services/owl-s/1.1/overview/>, Mär.2008.

- **[BBD+05] Web Service Modeling Ontology (WSMO).** W3C Member Submission, 2005.
J. d. Bruijn, C. Bussler, J. Domingue, D. Fensel, M. Hepp, U. Keller, M. Kifer, B. König-Ries, J. Kopecky, R. Lara, H. Lausen, E. Oren, A. Polleres, D. Roman, J. Scicluna und M. Stollberg.
<http://www.w3.org/Submission/2005/06/>, Nov. 2007.

- **[CaS06] Semantic Web Services, Processes and Applications.** Springer, 2006.
J. Cardoso und Amit P.Sheth.

- **[RLK05] Web Service Modeling Ontology (WSMO).** WSMO Final Draft, 2005.
D. Roman, H. Lausen und U. Keller.
<http://www.wsmo.org/TR/d2/v1.1/>, Apr. 2008.

- **[StH05] Semantic Web Services Tutorial.** DERI-Digital Enterprise Research Institute. 3rd International Conference on Web Services, 2005.
M. Stollberg und A. Haller.

- **[ABB+04] Semantic web services tutorial.** Technical report, International Semantic Web Conference 2004.
S. Arroyo, C. Bussler, J. de Bruijn, R. Lara, M. Moran, M. Stollberg, M. Zaremba, L. Vasiliu, M. Paolucci, K. Sycara, D. Martin, L. Cabral, J. Domingue.

- **[BBB+05] Semantic web services framework (SWSF).** W3C Member Submission, 2005.
S. Battle, A. Bernstein, H. Boley, B. Grosz, M. Gruninger, R. Hull, M. Kifer, D. Martin, D. L. McGuinness, S. McIlraith, G. Newton, D. De Roure, M. Skall, J. Su, S. Tabet, und H. Yoshida.
<http://www.w3.org/Submission/2005/07/>, Apr. 2008.

- **[FLP+07] Enabling Semantic Web Services.** Springer-Verlag Berlin Heidelberg, 2007
D. Fensel, H. Lausen, A. Polleres, M. Stollberg, D. Roman, J. Bruijn und J. Domingue.