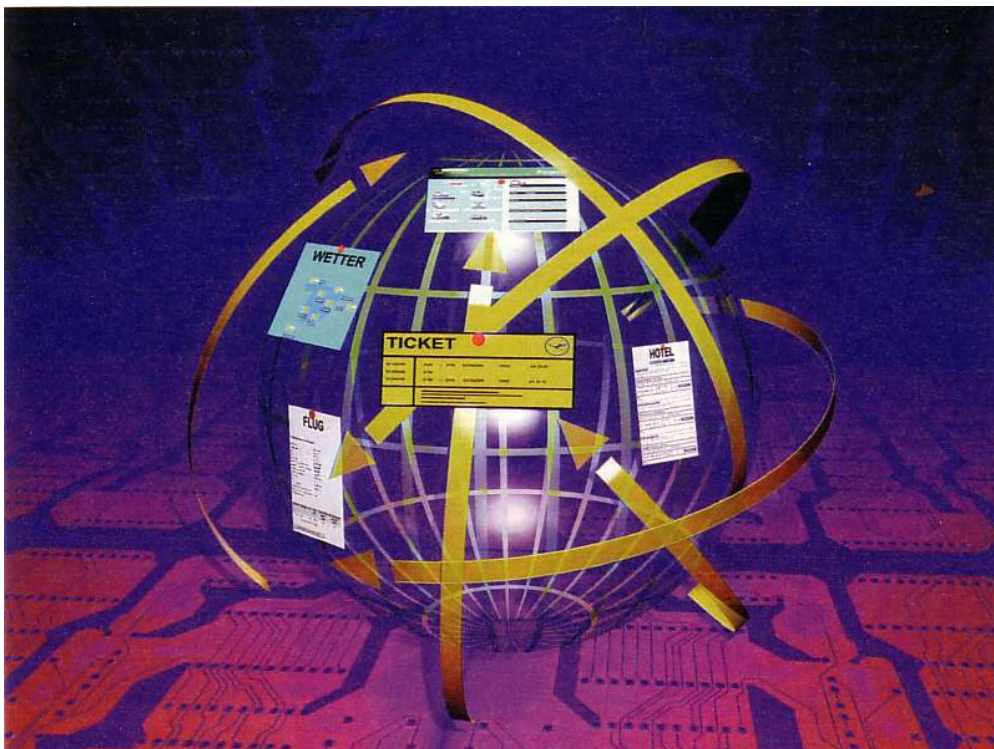


Semantic Web Services

Frank Bohdanowicz



Seminar Semantic Grid

AG Staab, FB4, Universität Koblenz
Januar 2005

Gliederung :

- 1) Was sind Web Services ?
- 2) Probleme heutiger Services im Web
- 3) Benötigte Technologien der Semantic Web Services
 - 1) Technologien des Semantic Web
 - 2) Technologien der Web Services
- 4) Semantic Matchmaking
- 5) Definition von Semantic Web Service Komponenten
- 6) Semantic Web Services Frameworks
- 7) Zusammenfassung
- 8) Literaturangaben

1) Was sind Web Services ?

Auf diese Frage gibt es mehrere Definitionen in unterschiedlicher Genauigkeit.

Allgemein sind Web Services Dienste, die über das Web erreichbar sind, wie auch der Begriff an sich schon sagt.

Genauer sind Web Services über das Internet erreichbare, durch Quellenidentifikation (Unified Resource Identification (URI)) eindeutig identifizierbare, spezielle Softwareprogramme mit offenen, zugänglichen Schnittstellen, für andere Softwareprogramme.

Noch genauer spezifiziert, sind Web Services entfernte Funktionsaufrufe (Remote Procedure Calls) über HTTP.

2) Probleme heutiger Services im Web

Heutige, im Web angebotene Services haben mehrere Probleme, welche bei zukünftigen Web Services gelöst sein sollen. Die Schnittstelle eines Großteils heutiger Services im Web sind dafür ausgelegt mit ein und demselben Programm zu kommunizieren. Oft handelt es sich dabei um Web-Browser, der aber nicht die gebotene Information der Services verarbeitet, sondern nur die Darstellung der Information, zB. über HTML-Tags. Der Web-Browser weiß nichts darüber, was zwischen den Tags <H1></H1> steht. Er weiß nur das er die Information die sich zwischen diesen Tagzeichen befindet besonders groß darstellen soll.

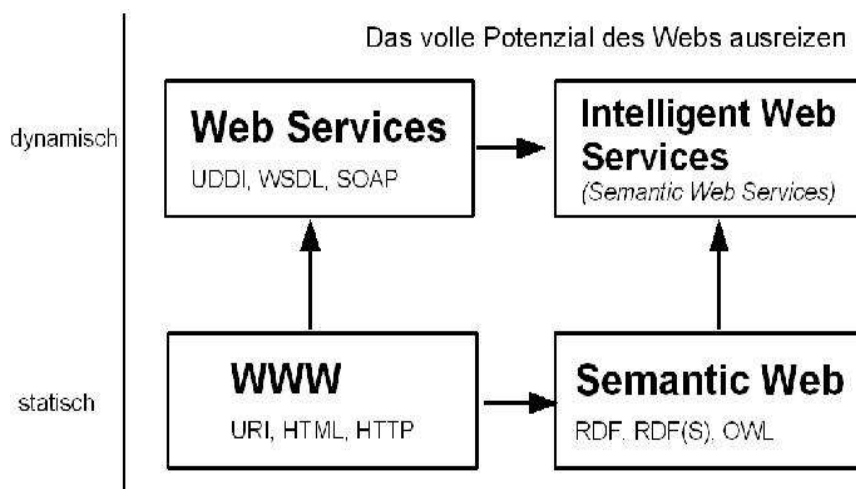
Viele Services im Web haben das Problem das sie nicht untereinander kombinierbar sind. Ein Service kann einen anderen Service meist nur mit erheblichem implementationstechnischen Aufwand nutzen, zB. in dem die Ausgabe des anderen Service „geparst“ wird und man auf gewisse Schlüsselwörter achtet und diesen Vorgang dann automatisiert. Da beinahe jeder Service im Web eine andere Ausgabe hat und diese sich dann oftmals auch verändert, kann das bei größeren Projekten zur wahren Sisyphus Arbeit ausarten.

Zum einen liegt das an den unterschiedlichen Standards, da sie von unterschiedlichen Firmen für unterschiedliche Zwecke entwickelt wurden, somit gibt es auch keine einheitlichen Schnittstellen. Services die das gleiche Angebot bieten können nicht oder nur unter viel Aufwand von ein und der selben Anwendung aufgerufen werden. Außerdem können sich Services im Web auch nicht einem anderen Computerprogramm beschreiben und somit können Computerprogramme Services auch nicht ohne weiteres finden, z.B. ohne die Interaktion durch den Menschen. All das soll durch zukünftige Web Services und auf Grundlage des Semantic Web funktionieren.

Dabei sollen Web Services mit standardisierte Schnittstellen in erster Linie

Zeit und Geld bei der Entwicklung von webgestützten Anwendungen sparen. Die Semantischen Web Services sollen dem Internet zusätzlich eine Automatisierung verleihen. So sollen Anwendungen auf den Heimrechnern selbständig Web Services finden, auswählen, aufrufen, mit ihnen interagieren, sie untereinander kombinieren und beobachten können. Dies soll durch sogenannte Agentenanwendungen möglich sein.

3) Technologien der Semantic Web Services



(Abbildung 1: Grundlagen der Semantischen Web Services (3))

Semantische Web Services sieht man auch als intelligente Web Services die einmal das volle Potenzial des Webs ausreizen sollen. Derzeit ist das World Wide Web eher noch eine statische, syntaxorientierte Sammlung von Informationen. Das World Wide Web heutiger Zeit legt mehr Wert auf die Darstellung der Information, als auf die Information an sich.

Das Semantic Web soll dem World Wide Web nun ermöglichen mit der Information selbst umgehen zu können, indem es der bereitgestellten Information eine Semantik gibt. Dabei soll das World Wide Web und dessen Techniken nicht abgelöst oder ersetzt werden. Das Semantic Web ist eher als eine Art Upgrade zu verstehen.

Web Services sind in erster Linie standardisierte Services, die unterschiedliche Informationen bereitstellen.

Die Semantischen Web Services gehen aus einer Zusammenarbeit der Web Services und des Semantic Webs hervor. Mit ihnen soll es praktisch möglich sein, Information von Computerprogrammen automatisch finden und verarbeiten zu lassen.

3.1) Die Technologien des Semantic Web

Das Semantic Web basiert auf den Ideen des WWW Erfinders Tim Berners-Lee und baut im wesentlichen auf den folgenden Techniken auf.

Zum einen RDF (Resource Description Framework), ein Schema zur Definition von Information im Internet. RDF liefert die Technologie zur Darstellung der Bedeutung von Begriffen und Konzepten in einer maschinenverarbeitbaren Form. RDF ist in erster Linie ein implementationsunabhängiges Schema, jedoch kann XML als Syntax benutzt werden und URIs zur Spezifizierung von Objekten, Konzepten, Eigenschaften und Beziehungen.

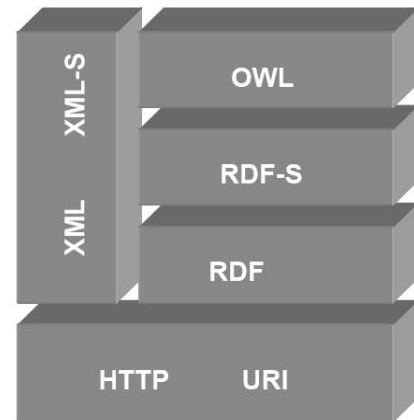


Abbildung 2: Semantic Web Technologien (4)

Auf RDF aufbauend ist das RDF-Schema (RDF-S), ein Vokabular zur Formulierung von Ontologien in RDF. Mit RDF-S bietet sich die Möglichkeit der Definition der Typisierung und Vererbung von Ressourcen und Eigenschaften. RDF-S entspricht objektorientiertem RDF.

Die letzte, hier vorgestellte Technologie des Semantic Web ist OWL (Web Ontology Language), eine Spezifikation des W3C um Ontologien anhand einer formalen Beschreibungssprache erstellen, publizieren und verteilen zu können. OWL basiert technisch auf dem oben genannten RDF und historisch auf DAML+OIL, der Vereinigung der DARPA Agent Markup Language und der Ontology Interchange Language.

3.2) Die Technologien der Web Services

Die Kommunikationsgrundlage der Web Services ist SOAP, was bis zur Version 1.1 für die Abkürzung von „Simple Object Access Protocol“ stand. Seit Version 1.2 ist SOAP kein Akronym mehr, da man nicht mehr nur auf Objekte zugreifen kann, sondern auf alles was irgendwie in XML strukturiert ist. SOAP ist eine Spezifikation des W3C für das Layout von Nachrichten, die einen einheitlichen Weg definiert um xml-kodierte Daten in dezentralen, verteilten Netzen zu versenden. SOAP baut auf HTTP als Kommunikationsprotokoll auf.

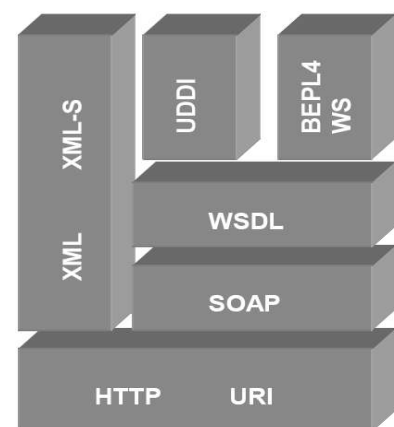


Abbildung 3: Web Service Technologien (4)

WSDL (Web Service Description Language) ist wie der Name schon sagt,

eine Sprache mit der man einen Web Service beschreiben kann. Es handelt sich hierbei um eine xml-basierte Sprache mit der man die Funktionalität von Web Services beschreibt. Mit einer WSDL-Datei hat man eine Beschreibung der Signatur von bereitgestellten Funktionen, deren Parameter und Rückgabewerte, der Web Services. Bislang ist WSDL nur als Entwurf beim W3C, allerdings können schon viele Entwicklungsumgebungen wie das „JDK 1.4“ oder „.Net“ mit WSDL umgehen.

Bei UDDI (Universal Description, Discovery and Integration) handelt es sich im wesentlichen um einen globalen Verzeichnisdienst im Web, der Daten und Serviceangebote von Unternehmen enthält. UDDI entspricht in etwa den bekannten „gelben Seiten“. Erweitert man UDDI mit einer URI auf eine WSDL-Datei, so soll man die verzeichneten Web Services nicht nur finden können. Die Agentenapplikation soll über die WSDL-Datei den Web Service dann auch automatisch aufrufen und mit ihm interagieren. Da aber normalerweise keine explizite semantische Information im UDDI definiert ist, limitiert sich auch die automatische Einbeziehung der WSDL Beschreibung auf Fälle, in denen sich Anbieter und Anfrager vorher auf gemeinsame Ontologien, Protokolle und Wissen über die Geschäftsoperationen verständigt haben.

UDDI ist eine von Ariba, Microsoft und IBM ins Leben gerufene Initiative.

BPEL4WS (Business Process Execution Language for Web Services) ist eine xml – basierte Sprache zur Beschreibung des Verhaltens von Geschäftsprozessen von Web Services und deren Kompositionen. BPEL4WS ist eine Standard-Technologie um verschiedene Web Services miteinander zu verlinken und zu einem Business Prozess zusammenzufassen. Es baut auf Microsofts Xlang und IBMs WSFL auf, die selbst untereinander nicht kompatibel sind.

Damit wären alle wichtigen Technologien, die Semantic Web Services möglich machen, vorgestellt. In dem folgenden Schichtenmodell wird gezeigt wie die einzelnen Technologien zusammenhängen und aufeinander aufbauen.

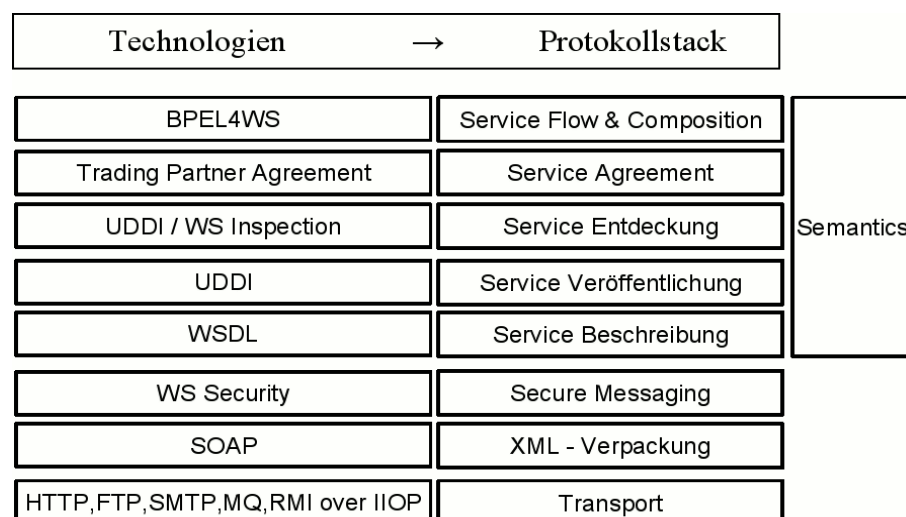
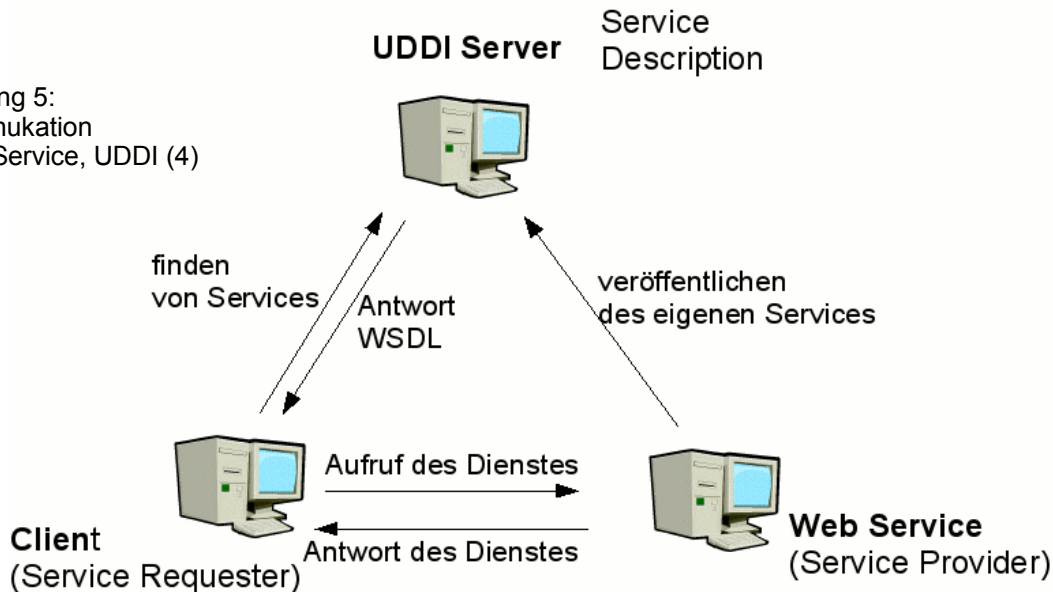


Abbildung 4:
Schichtenmodell
Semantic Web
Service (3)

Unter Verwendung dieser Technologien könnte ein Beispiel so aussehen:

Ein Programm fragt in einem UDDI-Verzeichnis nach Web Services, die ein bestimmtes Produkt anbieten. Von diesen Web Services wählt es den passenden aus und holt sich die WSDL-Beschreibung, die angibt, wie der Service zu verwenden ist. Über SOAP-Nachrichten kann das Programm nun mit dem Web Service kommunizieren, seine Dienste in die eigene Programmstruktur einbinden und die Angebote für das Produkt direkt weiterverarbeiten.

Abbildung 5:
Kommunikation
Client, Service, UDDI (4)



4) Semantic Matchmaking

Semantische Web Services sollen automatisch auffindbar, aufrufbar und auch untereinander kombinierbar sein. Hier ist es notwendig dass die gesuchten Services im UDDI gefunden werden. Dabei hilft das semantische Matchmaking, das auf Ontologien aufsetzt und einen passenden Service zu einer Suchanfrage führen soll. Man unterscheidet hier 4 Stufen der Übereinstimmung.

exact – wenn die Anfrage äquivalent zu einem Eintrag im UDDI ist und durch diesen Service komplett erfüllt werden kann

plugin – wenn die Anfrage von einem Eintrag im UDDI mit abgedeckt wird und sie der Service unter anderem erfüllt

subsumes – wenn die Anfrage von einem UDDI Eintrag nur zum Teil abgedeckt wird und der dahinterstehende Service nur ein Teilziel erfüllt

fail – wenn die Anfrage von diesem Service nicht erfüllt werden kann

5) Definition von Semantic Web Service Komponenten

Semantische Web Services können in 3 Dimensionen aufgeteilt werden. Aktivitäten, Architektur und Service Ontologie.

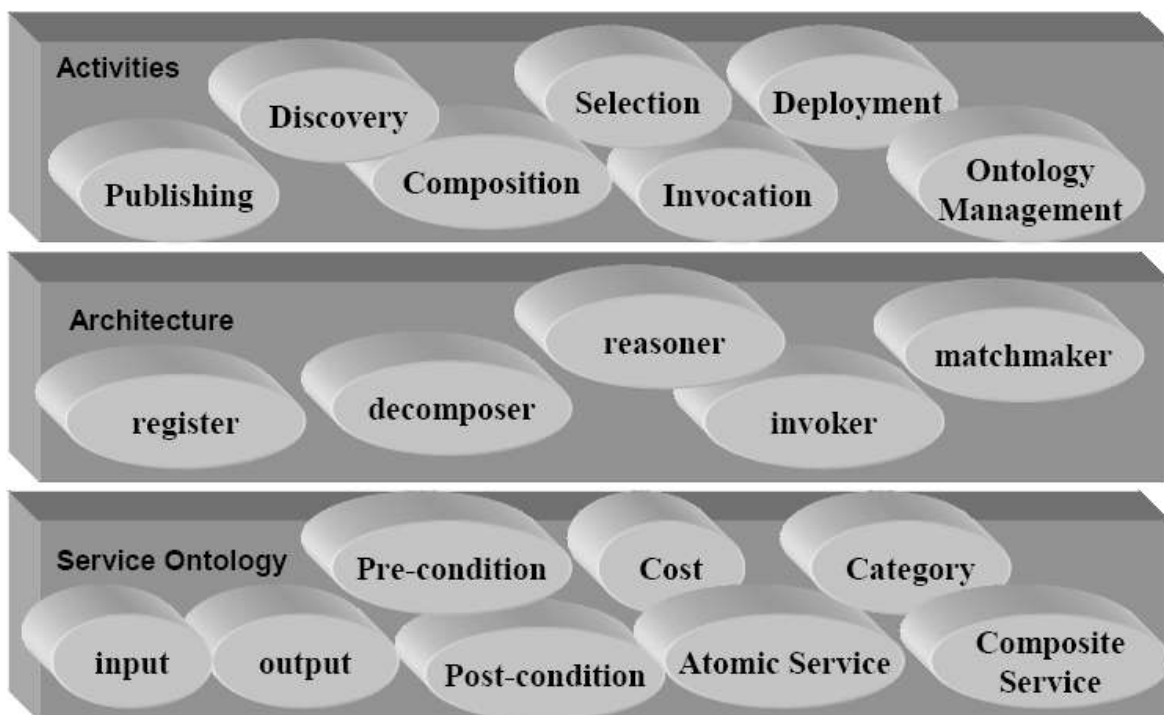


Abbildung 6: Komponenten der Semantic Web Services (4)

Die Semantic Web Service Aktivitäten definieren die funktionalen Ansprüche die man an den Web Service hat.

Publishing – Wie wird er veröffentlicht

Discovery – Wie kann er von anderen Anwendungen entdeckt werden

Composition – Greift er auf andere Web Services zu, oder besteht er gar komplett aus einer Komposition einzelner Web Services

Selection – Wie kann er von anderen Anwendungen ausgewählt werden.

Invocation – Wie kann er von anderen Anwendungen aufgerufen werden

Deployment – Wie wird er entwickelt bzw. implementiert

Ontology Management – Wie greift der Web Service auf Ontologien zu

Die Semantic Web Service Architektur definiert die Komponenten, die notwendig sind um die Aktivitäten bewältigen zu können.

register – liefert den Mechanismus für die Veröffentlichung und Lokalisierung

decomposer – wird benötigt um komponierte Web Services auszuführen

invoker – vermittelt zwischen Service Anfrager und Provider um einen Service aufzurufen

reasoner – liefert den Entscheider der die Interpretation von semantischen Beschreibungen und Anfragen unterstützt

matchmaker – vermittelt zwischen Anfrager und Register während der Suche und Auswahl eines Web Service

Die Semantic Web Service Service Ontology definiert den Web Service selbst, was er macht und wie er was macht.

Input – Die Definition der Eingabe

Output – Die Definition der Ausgabe

Pre-condition – Die Definition der Vorbedingungen

Post-condition – Die Definition der Nachbedingungen

Cost – Die Definition der Kosten

Atomic Service – Definition der atomaren Web Services

Composite Service – Definition der komponierten Web Services

Category – Definition der Kategorie, die der Service abdeckt

6) Semantic Web Services Frameworks

Frameworks dienen der Modellierung von Semantischen Web Services. Es gibt mehrere verschiedene Frameworks, hier werden allerdings nur 2 behandelt.

Zum einen OWL-S (Web Ontology Language for Services) was oft als Quasi-Standard gesehen wird und zum anderen WSMF (Web Service Modeling Framework).

OWL-S geht historisch aus DAML-S (DARPA Agent Markup Language for Services) hervor. OWL-S selbst ist die Ontologie, mit der Web Services beschrieben werden.

OWL-S definiert eine Reihe von Klassen und Eigenschaften zur Beschreibung eines Web Services.

Service Profile – dienen der Beschreibung des Services zum Zwecke des Entdeckens

Service Modelle – stellen eine semantische Beschreibung und Charakterisierung von Web Services

Service Grounding – beinhalten die Bindung an physikalische Ressourcen und implementationsspezifische Details sowie die Art des Nachrichtenübertragungsformats.

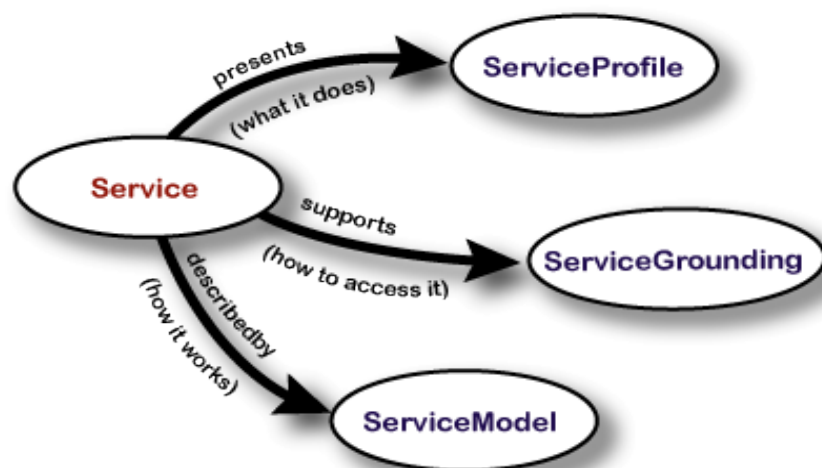


Abbildung 7:

OWL-S
Service Modell (1)

OWL-S unterteilt Web Services Prozesse in 3 Gruppen

Atomic Process – ist ein direkt aufrufbarer Prozess, ohne Unterprozesse, der nur eine spezielle Aufgabe in einem Schritt erledigen kann.

Simple Process – ist eine abstrakte Sicht eines Prozesses, zB. wie man mehrere atomic Processes zusammenfügen kann. Ein Simple Process selbst ist nicht aufrufbar.

Composite Process – ist ein Prozess der aus mehreren anderen Prozessen mittels Konstrukten wie sequence, split, join, any order oder choice zusammengefügt wurde

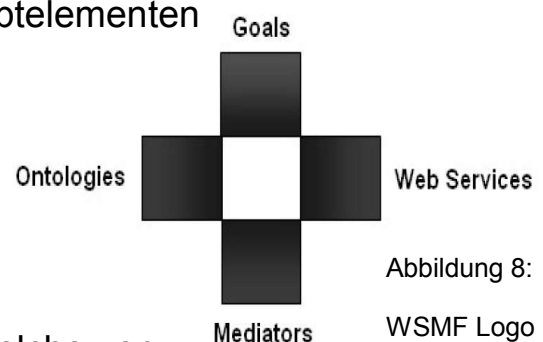
WSMF, das zweite hier vorgestellte Framework, hat als Hauptziel die volle Unterstützung von E-Commerce durch die Anwendung von Semantic Web Technologie bei Web Services. Hier spezialisiert man sich auf sogenannte Semantic Web enabled Web Services (SWWS) die Objektiv gesehen das gleiche wie Semantic Web Services liefern, nur mehr Wert auf das vorhanden sein des Semantic Web legen. SWWS will das Web als statische Sammlung von Informationen, in eine verteilte Berechnungsmaschine auf Basis des Semantic Web transformieren.

WSMF zentriert sich auf 2 komplementäre Prinzipien

- eine starke Entkopplung der unterschiedlichen Komponenten die E-Commerce realisieren
- ein Vermittlungsservice, basierend auf der Semantic Web Technologie, mit dem man unterschiedlich stark kommunizieren kann.

Die zugrundeliegende Ontologie von WSMF ist WSMO (Web Service Modeling Ontology), welche auf F-Logik basiert.

WSMF besteht aus 4 unterschiedlichen Hauptelementen



Ontologien – welche die Terminologie, welche von den Web Services genutzt wird, bereitstellen

Ziele – welche die zu lösenden Probleme der Web Services definieren

Vermittler – die interoperationale Probleme überbrücken

Web Services – Beschreibungen, die verschiedene Aspekte des Web Service definieren

Die WSMO Service Ontologien beinhaltet Definitionen für Ziele (Goals), Vermittler (Mediators) und Web Services.

Eine charakteristische Eigenschaft der WSMO Architektur ist, dass Ziele, Web Services und Ontologien durch 4 Typen von Vermittlern verknüpft werden.

OO – Vermittler die Ontologien mit Ontologien verknüpfen

WW – Vermittler die Web Services mit Web Services verknüpfen

WG – Vermittler die Web Services mit Zielen verknüpfen

GG – Vermittler die Ziele mit Zielen verknüpfen

WSMO und OWL-S teilen die Ansicht, dass Ontologien essentiell sind für die automatische Interaktionen mit und zwischen Web Services.

OWL-S geht es aber eher um eine Spezialisierung und Generalisierung bei der Modellierung, also unabhängige, atomare Prozesse, die je nach Bedarf zu größeren Prozessen zusammengeschlossen werden können.

WSMO strebt eher gleichberechtigte „Insellösungen“ an, die über skalierbare Vermittler unterschiedlich stark miteinander kommunizieren können.

Wie oben schon erwähnt, wird WSMF von SWWS genutzt, das hier kurz vorgestellt werden soll.

Die SWWS Architektur lässt sich in 3 Bereiche unterteilen.

Frontend-Werkzeuge – diese ermöglichen die Modellierung, Entwicklung und das Simulieren von semantisch-basierten Web Services, sowie die Verwaltung und Handhabung des Gesamtsystems

Kern-Komponenten – liefern die benötigten Funktionen zur Realisierung der Web Services

Speicher und externe Komponenten – liefern die Schnittstellen, die den Zugriff auf externe Daten und Engines ermöglichen zB. auf Ontologien-Server

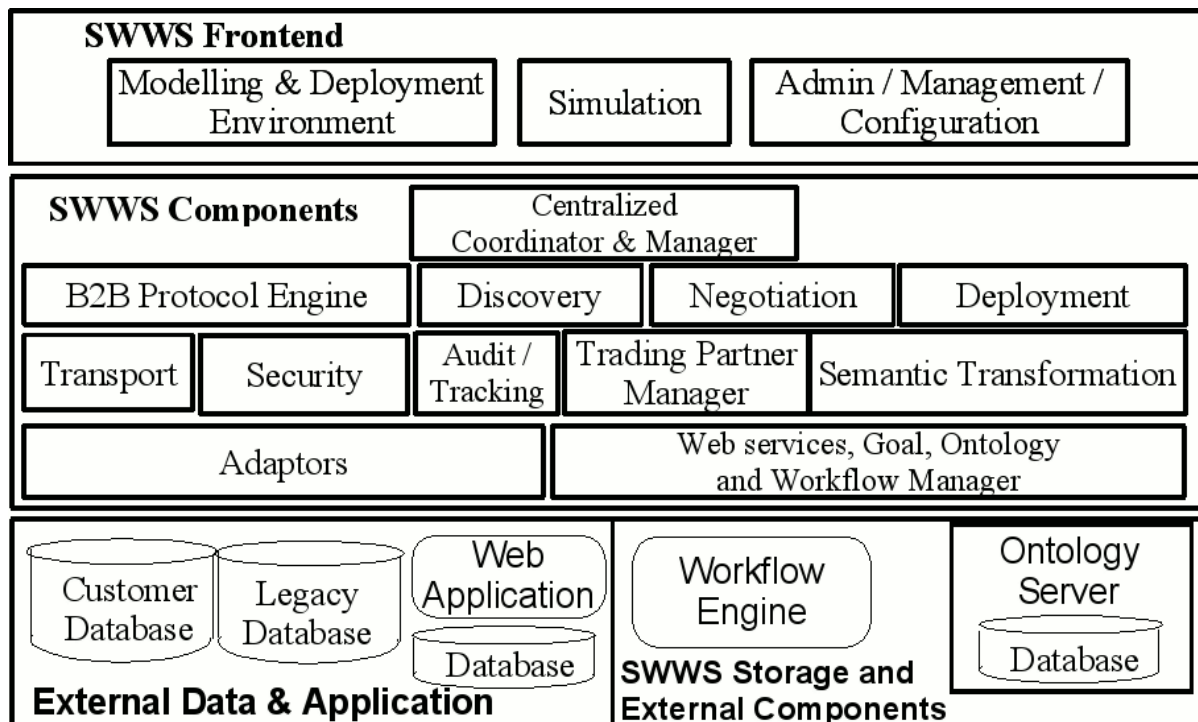


Abbildung 9: SWWS Architektur (3)

Die Kern-Komponenten werden hier noch im Einzelnen vorgestellt.

- Centralized Coordinator & Manager
ist die Zentrale, die alle anderen Komponenten koordiniert und managed.
- B2B Protocol Engine
hier sind die Nachrichtensequenzen definiert
- Discovery
diese Komponente ist für die Entdeckung des Web Service zuständig
- Negotiation
liefert eine semantische Unterstützung für Geschäftsverhandlungen, hier findet auch Semantic Matchmaking statt
- Deployment
unterstützt eine einfache Entwicklung des Modellierten Web Service auf unterschiedlichen Plattformen wie zB. J2EE oder .Net
- Transport
hier ist das Netzwerk und seine Protokolle wie wie HTTP, FTP, usw definiert
- Security
dient der Authentifizierung und Verschlüsselung
- Audit/Tracking
dient der Speicherung von Geschäftsprozessen um sie im Nachhinein nachweisen oder prüfen zu können
- Trading Partner Manager
hier wird ein Profil des Handlungspartners gespeichert, Name, Adresse, usw. sowie Protokolle, auf die man sich für die Verständigung geeinigt hat.

- Semantic Transformation
hier wird die Transformation von unterschiedlichen Formaten geregelt, z.B. von einem Sendeformaten in ein Anderes.
- Adapter
machen es möglich Programme zu kombinieren trotz unterschiedlichen Outputs und Inputs.
- Web Services, Goal, Ontology and Workflow Manager
liefert die Api und ermöglicht somit standardisierte Schnittstellen und vereinfachte Zugriffsmöglichkeiten für Programme.

7) Zusammenfassung

Zusammenfassend kann man sagen, dass das Semantic Web die vorhandene Information durch Ontologien aufwerten und maschinenlesbar machen wird und die Semantic Web Services diese Information bereitstellen und verknüpfen können, wodurch ein verbesserter Zugriff auf die Daten entsteht.

Zukünftige Softwareanwendungen werden auch einen Großteil Ihres Potenzials ins Internet auslagern und lediglich bei Bedarf in die eigene Struktur einbinden, womit sie wesentlich aktueller, mächtiger und flexibler sein werden, als es heutigen Softwareanwendungen möglich ist.

Literatur :

- 1) OWL-S: Semantic Markup for Web Services
<http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122>
- 2) D. Fensel and C. Bussler:
The Web Service Modeling Framework WSMF,
Electronic Commerce: Research and Applications, 1 (2002) 113-137
<http://www.wsmo.org/publinks.html>
- 3) Christoph Bussler, Dieter Fensel, Alexander Maedche
A Conceptual Architecture for Semantic Web Enabled Web Services
- 4) Liliana Cabral, John Domingue, Enrico Motta, Terry Payne
and Farshad Hakimpour
Approaches to Semantic Web Services: an Overview and Comparison
- 5) Semantic Web Services, Semantic Web enabled Web Services
IST Projekt Präsentation
- 6) c't magazin für computer technik, 6/2002, S 132ff, S 236ff
(u.a. Titelbild)
- 7) Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, Katia Sycara
Semantic Matching of Web Services Capabilities