

Semantic Web Services – Realisierung der SOA Vision mit semantischen Technologien

Michael Stollberg, Martin Hepp, und Dieter Fensel
Digital Enterprise Research Institute (DERI)
Leopold-Franzens Universität Innsbruck
Technikerstrasse 21a, 6020 Innsbruck, Austria
{michael.stollberg, martin.hepp, dieter.fensel}@deri.org

Abstract

Das Ziel service-orientierter Architekturen (SOA) ist das dynamische Finden und Ausführen von Web Services, um eine konkrete Nutzeranfrage zu lösen. Dazu ist die Automation des Web Service Nutzungsprozesses nötig. Dies ist mit den derzeitigen Technologien (WSDL, SOAP, UDDI) nicht möglich, da diese eine manuelle Inspektion existenter Web Services hinsichtlich ihrer Benutzbarkeit erfordern. In diesem Artikel stellen wir das Konzept von Semantic Web Services sowie den derzeitigen Stand der Forschung vor, welche inferenz-basierte Technologien zum automatischen Finden, Komponieren, und Ausführen von Web Services bereitstellen.

1. Einleitung

Service-Orientierte Architekturen (SOA) sind in aller Munde als neues Paradigma für IT Systemdesign. Die Idee ist, dynamisch Web Services zu finden und auszuführen, um eine konkrete Nutzeranfrage zu lösen. Damit sollen die Probleme monolithischer Systeme hinsichtlich Interoperabilität und funktionaler Spezialisierung gelöst werden. Doch die Realisierung dieser Vision wirft einige Probleme auf: wie sollen jene Web Services gefunden werden, mit denen eine spezifische Anfrage gelöst werden kann? Wie können potenziell auftretende Heterogenitäten behandelt werden? Wie kann man SOA Technologie automatisieren, um sie wirklich nutzbar zu machen?

Im Folgenden stellen wir *Semantic Web Services* als einen Lösungsansatz dafür vor [5]. Basierend auf semantischen Beschreibungen werden Web Services automatisch gefunden, kombiniert, und ausgeführt. Die Grundlage stellen Ontologien dar – Wissensmodelle, die bedeutungserhaltende Informationsverarbeitung ermöglichen. Damit werden zwei parallele Entwicklungen zur besseren Nutzung des World Wide Web als weltweite Informations- und Kommunikationsplattform integriert: das *Semantic Web* zur verbesserten Verarbeitung von Web Inhalten auf der Grundlage strukturierter und bedeutungsbeschreibender Markup-Sprachen, und *Web Services* zur web-basierten Bereitstellung und Nutzung von Softwarekomponenten.

2. Die SOA Vision und Web Services

Die grundlegende Idee von SOA ist, dass aus einer Vielzahl angebotener Web Services jene dynamisch gefunden, kombiniert, und ausgeführt werden, die zur Lösung einer spezifischen Nutzeranfrage benötigt werden. Dadurch sollen mehrere Probleme existenter Systemarchitekturen gelöst werden:

- Wiederverwendbarkeit programmatischer Ressourcen
- Entkopplung interner und externer Prozesse bei Anbietern und Konsumenten
- das Integrationsproblems
- Plattformunabhängigkeit durch Nutzung des Web als Infrastruktur für Kommunikation und Informationsaustausch.

2.1. Web Services

Das Konzept von Web Services wurde Ende der 1990er Jahre auf Initiative führender Softwarekonzerne entwickelt. Ein Web Service ist ein Programm, welches durch ein Interface über das Web aufgerufen werden kann. Dies hat zwei Vorteile hinsichtlich der Zielsetzung von SOA: (1) die konkrete Implementierung des Web Service ist unabhängig vom Interface, und (2) die Nutzung des Internet als bestehende Infrastruktur zum Informationsaustausch zwischen Programmen. Daher werden Web Services als technische Lösung des Integrationsproblems proklamiert: ein Anbieter kann seine Software intern entwickeln und verwalten, während die Nutzung und Interaktion mit Endnutzern sowie Geschäftspartnern über das Interface geschieht [1].

Die Interoperabilität soll durch die Standardisierung der Web Services Interfaces sichergestellt werden. Dazu wurden um die Jahrtausendwende erste Versionen von Web Service Technologien durch internationale Standardisierungsgremien verabschiedet.

1. WSDL – The Web Service Description Language (W3C, <http://www.w3.org/TR/wsdl>)

Eine XML-basierte Sprache zur Beschreibung eines Web Service Interface. Im Prinzip werden die Messages definiert, die der Nutzer mit dem Web Service austauschen kann, um dessen Funktionalität zu konsumieren. Des Weiteren werden der „endpoint“ (die Web-Adresse des Web Service) sowie die verwandten XML-Schema definiert.

2. SOAP – Simple Object Access Protocol (W3C, <http://www.w3.org/TR/soap/>)

Eine Technologie zum Austausch von XML Daten über das Web. Darüber soll der Informationsaustausch zwischen dem Web Service Anbieter und dem Konsumenten stattfinden.

3. UDDI – Universal Description, Discovery, and Integration (OASIS, www.uddi.org)

Eine Technologie zur Registrierung, Verwaltung, und Suche von Web Services. Darin können Web Services hinsichtlich verschiedener Aspekte beschrieben werden (Anbieter, Keywords bzgl. der angebotenen Funktionalität); Nutzer durchsucht dieses, um geeignete Web Services zu finden.

Abbildung 1 illustriert den groben Ablauf für das Konsumieren eines Web Service mittels dieser Technologien. Der Nutzer durchsucht zunächst die UDDI Registry, und wählt den zu nutzenden Web Service aus. Aus der zugehörigen WSDL Beschreibung können die detaillierten Informationen zum Aufrufen des Web Services ersehen werden. Dann wird der Web Service aufgerufen, indem die entsprechenden Informationen per SOAP ausgetauscht werden.

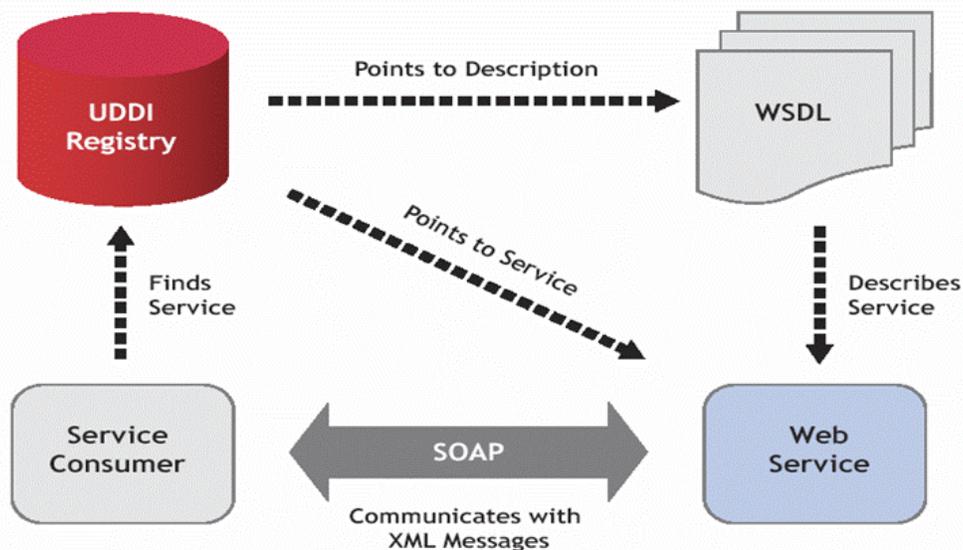


Abbildung 1: Web Service Technologien

2.2. Unzulänglichkeiten von Web Service Technologien

Mit diesen Technologien kann man also prinzipiell Web Services bereitstellen, finden, und ausführen. Allerdings ist dies noch sehr weit entfernt von der Realisierung der SOA Vision. Der einzige automatisierte Prozess ist die konkrete Ausführung eines Web Services durch den Austausch von SOAP Messages. Die gesamte vorhergehende Nutzbarkeitsanalyse – das Wesentliche zur

wirklichen Nutzung von Web Services im Sinne von SOA – erfordert menschliche Intervention. Dieses ist nicht trivial. Insbesondere treten dabei folgende Schwierigkeiten auf:

1. Auffinden des richtigen Web Service

Zunächst muss man den richtigen Web Service finden. Dabei ist zum einen eine sehr große Menge von Web Services zu erwarten – mehrere Millionen, wenn man als grobe Größenordnung zu Grunde legt, dass jede derzeit registrierte .com-domain mindestens einen Web Service anbietet. Zu anderen treten die funktionalen Unterschiede gleichartiger Web Services zumeist im Detail auf – zum Beispiel die unterschiedlichen Lieferkonditionen von Transport- und Logistikanbietern.

2. Feststellen der Nutzbarkeit

Nachdem der richtige Web Service gefunden ist, muss der Nutzer die richtigen Informationen in der richtigen Reihenfolge bereitstellen können, um den Web Service aufzurufen. Die relevanten Anforderungen stehen in der WSDL Beschreibung – allerdings sind darin lediglich die möglichen Messages auf syntaktischer Ebene beschrieben. Daher muss der Nutzer diese manuell analysieren und dann ein kompatibles Gegenstück dazu bereitstellen, um den Web Service zur Lösung der spezifischen Anfrage nutzen zu können.

3. Behandlung von Heterogenitäten

Gemeinhin bekannt als das Integrationsproblem, können Heterogenitäten auf verschiedenen Ebenen die erfolgreiche Nutzung eines Web Services verhindern. Dies können unterschiedliche Datenformate oder –modelle sein, sowie Unverträglichkeiten zwischen den öffentlichen Geschäftsprozessen des Anbieters und des Konsumenten. Insbesondere im Umfeld des Internet stellen solche Heterogenitäten eher den Normalfall dar.

Kurzum: man braucht ausdrucksstärkere Beschreibungen von Web Services sowie geeignete Mechanismen, um die Nutzbarkeitsanalyse von Web Services besser zu unterstützen und zu automatisieren. Nur damit lässt sich die SOA Vision wirklich realisieren.

3. Semantische Beschreibung von Web Services

Der Ansatz von Semantic Web Services setzt sich genau dies zum Ziel. Es werden detaillierte Beschreibungen der angebotenen Funktionalität sowie des Interfaces zur Nutzung eines Web Service definiert. Diese Beschreibungen basieren auf Ontologien, also in logischen Sprachen spezifizierten, bedeutungsdefinierenden Wissensmodellen. Auf Grundlage dieser semantischen Beschreibungen dienen inferenz-basierte Mechanismen zum Auffinden, Kombinieren, und Ausführen von geeigneten Web Services. Damit soll zum einen eine weitgehende Mechanisierung der Nutzenanalyse von Web Services erreicht werden; zum anderen soll das Integrationsproblem als integraler Bestandteil semantischer SOA Technologie durch entsprechende Ontologie-Techniken behandelt werden.

3.1. Ontologien – Semantic Web

Parallel zu Web Services proklamierte Tim Berners-Lee (Erfinder des World Wide Web und Direktor des W3C) das *Semantic Web* [2]. Als Erweiterung des bestehenden Web sollen auf der Basis strukturierter und ausdrucksstärkerer Mark-up Sprachen die Verarbeitung von Web Inhalten verbessert und automatisiert werden. Abbildung 2 zeigt den im Jahr 2005 revidierten „Semantic

Web Layer Cake“: die Basis bilden die existenten Web Technologien (URI, Unicode, XML, Namespaces); darauf folgt die Ontologie Ebene (RDF, OWL, Rules, und SparQL), welche derzeit das Hauptaugenmerk der Standardisierungsaktivitäten bildet. Darüber sollen ein Logik Framework folgen, vertikal gekoppelt mit sicherheitsrelevanten Aspekten.¹

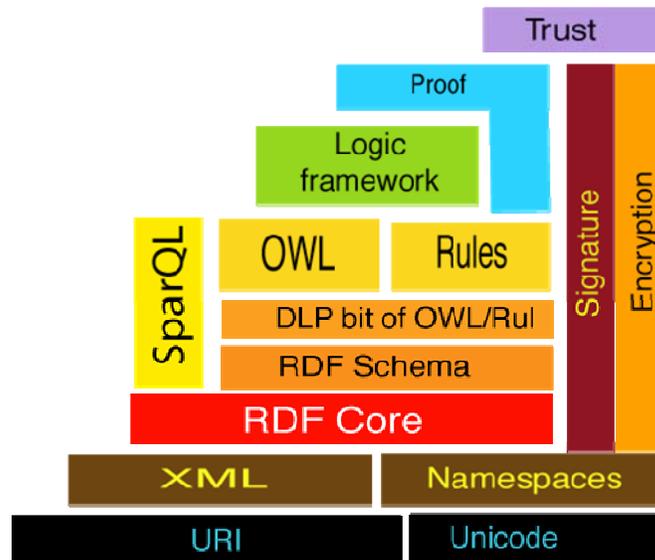


Abbildung 2: Semantic Web Layer Cake

Die zentrale Basis für das Semantic Web sind Ontologien [4]. In Anlehnung an die gleichnamige philosophische Disziplin ist dies eine moderne, weithin akzeptierte Wissensmodellierungstechnik in der Künstlichen Intelligenz (KI). Darin ist eine Ontologie als „explizite Formalisierung eines geteilten konzeptionellen Modells“ definiert. Also: ein konzeptionelles Modell einer Wissensdomäne, welche von allen Beteiligten akzeptiert wird und damit die Basis eines gemeinschaftlichen Verständnisses zum bedeutungserhaltenden Informationsaustausch bildet. Dabei soll implizites Wissen so weit wie möglich externalisiert werden, so dass widersprüchliche Interpretationen vermieden werden. Dieses Modell wird dann in einer geeigneten logischen, maschinenlesbaren Sprache formalisiert (jene aus der Ontology Ebene, s.o.).

Die Verwendung von Ontologien als Daten-, bzw. Wissensmodelle für das Semantic Web bietet folgende Vorteile – vor allem im Vergleich zu den existenten, syntaktischen Web Sprachen:

1. bedeutungsdefinierende Beschreibungen der Welt oder einer Domäne (im Gegensatz zu technischen Datenmodellen wie z.B. in relationalen Datenbankschemata)
2. aussagekräftige und bedeutungserhaltende Informationsverarbeitung durch KI Reasoning
3. Behandlung von Heterogenitäten auf der semantischen Ebene durch „Ontology Mapping“ (siehe unten) .

¹ siehe die W3C Semantic Web Activity, homepage: <http://www.w3.org/2001/sw/>

3.2. Annotation von Web Services

Der Sinn der erweiterten Beschreibung von Web Services ist es, die Funktionalität, das Interface, sowie weitere Aspekte eines Web Services derart hinreichend zu beschreiben, dass die Nutzbarkeitsanalyse automatisiert werden kann. Ontologien sowie andere formale Sprachen dienen als Basis, so dass wir von einer semantischen Annotation eines Web Service sprechen.

Abbildung 3 zeigt die Erweiterung von syntaktischen hin zu semantischen Beschreibungen von Web Services. Die wesentlichen Aspekte dabei sind:

- 1) **Ontologien als Datenmodell:** sowohl jedes Beschreibungselement als auch alle Daten, die zwischen einem Web Service und dessen Nutzer ausgetauscht werden, sind einer Ontologie zugeordnet. Damit werden die Vorteile von Ontologien zur semantischen Informationsverarbeitung genutzt, und die Integration mit dem Semantic Web gesichert.
- 2) **formale, explizite Funktionalitätsbeschreibung:** als neuer Aspekt wird die von einem Web Service angebotene Funktionalität explizit beschrieben. Formal wird dies durch *preconditions* (Bedingungen vor dem Aufrufen des Web Service) und *effects* (Bedingungen nach erfolgreicher Ausführung) definiert. Diese Beschreibung ist vor allem für das Auffinden der richtigen Web Services relevant.
- 3) **formale Interface Beschreibung:** zur Automatisierung der Nutzbarkeitsfeststellung wird das Web Service Interface reicher beschrieben. In Ergänzung zu den Informationen in der WSDL Beschreibung werden vor allem die möglichen Abfolgen des Informationsaustausch in formalen Prozesssprachen definiert. Außerdem beschreibt ein weiteres Interface das Verhalten eines Web Services zur Interaktion mit anderen Web Services, so dass komplexere Web Services als Kompositionen definiert werden können.
- 4) **nicht-funktionale Aspekte:** diese umfassen Informationen über den Anbieter sowie quality-of-service Aspekte (Sicherheit, Erreichbarkeit & Stabilität, örtliche Bezogenheiten, etc.)

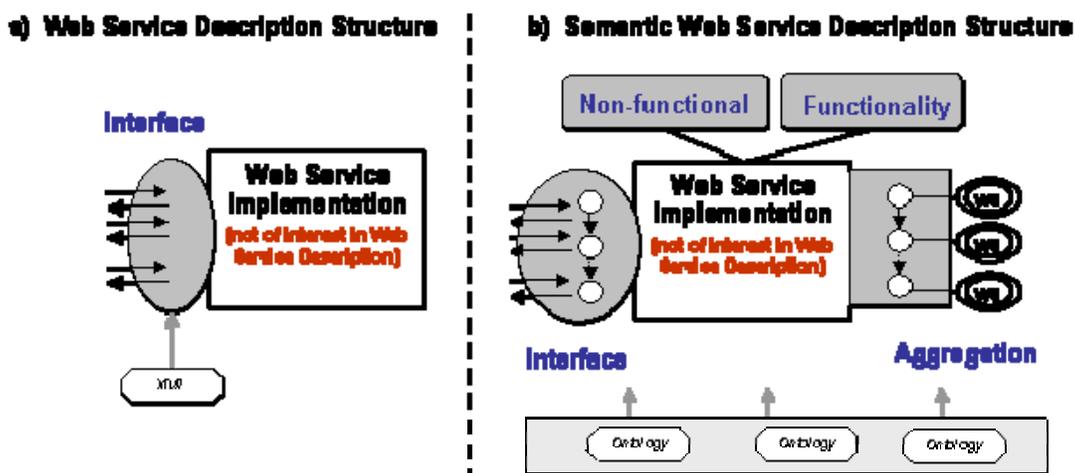


Abbildung 3: Semantische Annotation von Web Services

4. Semantische Web Service Technologien

Auf der Grundlage derartig annotierter Web Services sowie von Ontologien können intelligente, inferenzbasierte Mechanismen zur Automatisierung der kompletten Nutzbarkeitsanalyse und Ausführung von Web Services definiert werden.

Wie in der SOA Vision angedacht, werden für eine konkrete Anfrage die nutzbaren Web Services automatisch gefunden und ausgeführt. Abbildung 4 zeigt den Verarbeitungsprozess sowie die wesentlichen Inferenzmechanismen dafür. Zuerst werden die potenziell nutzbaren Web Services gesucht; dies geschieht durch semantisches matchmaking hinsichtlich der funktionalen Beschreibungen (discovery). Sollte kein direkt nutzbarer Web Service gefunden werden, wird eine Kombination mehrerer Web Services erstellt (composition). Als nächstes werden die gefundenen Web Services und Kompositionen im Hinblick auf nicht-funktionalen Aspekte gewichtet und selektiert (selection). Als letzter Schritt der Nutzbarkeitsanalyse wird die Aufrufbarkeit im Hinblick auf das Interface geprüft, also ob der Konsument den vom Web Service angebotenen Kommunikationsprozess unterstützen kann (behavioral conformance). Wenn dies erfolgreich beendet ist, wird der Informationsaustausch zwischen Web Service und Anfrager durchgeführt (execution).

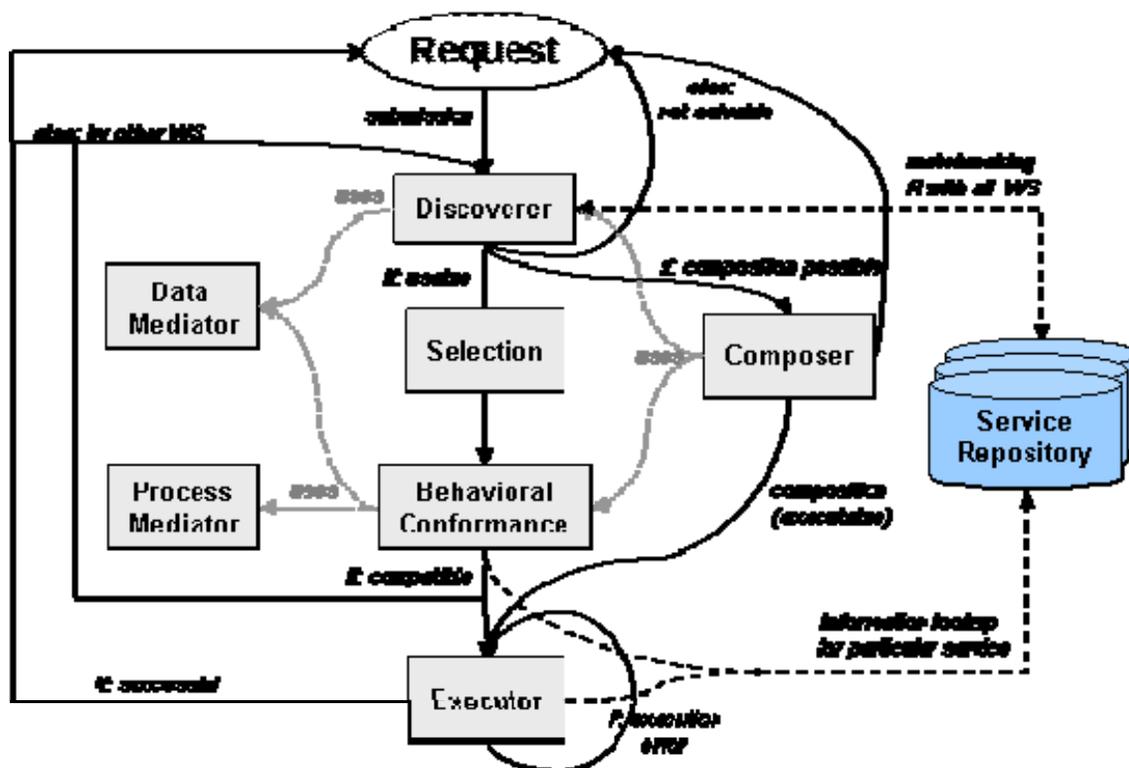


Abbildung 4: Automatisierte Nutzung von Web Services

Im Hinblick auf das Integrationsproblem werden zusätzlich so genannte *Mediatoren* genutzt. Diese stellen Mechanismen zur Verfügung, die möglicherweise auftretende Heterogenitäten auflösen und behandeln können. Dabei werden die Heterogenitäten auf semantischer Ebene analysiert, also auf struktureller Grundlage der formalen Beschreibung und nicht fallspezifisch für eine konkrete Anfrage. Für Web Services sind dabei zwei Arten von Mediationstechniken von zentraler Bedeutung:

- **Data Level Mediation:** dies bezieht sich auf Heterogenitäten, die auf Grund der Benutzung unterschiedlicher Datenformate und –modelle durch Web Service Anbietern und Nutzer entstehen. Erstere werden durch *Adapater* gelöst, die zwischen Datenformaten konvertieren. Wichtiger und komplizierter handzuhaben sind konzeptionelle Unterschiede, welche durch die Verwendung unterschiedlicher Ontologien entstehen. Zu deren Behandlung werden *semantische Integrationstechniken* verwandt – einer der wesentlichen Vorteile von Ontologien als Wissensrepräsentationstechnik. Das Herzstück dabei sind *ontology mappings*, welche eine semantische Brücke zwischen heterogenen Ontologien definieren – zum Beispiel, dass alle „Erwachsenen“ aus der Ontologie A identisch sind mit „alle Personen über 18 Jahre“ aus der Ontologie B. Damit kann eine bedeutungserhaltende Interaktion gewährleistet werden.
- **Process Level Mediation:** dies bezieht sich auf Heterogenitäten zwischen den öffentlichen Geschäftsprozessen von Web Service Anbietern und Nutzern. Diese manifestieren sich in Inkompatibilitäten der Interfaces von Web Services, die interagieren sollen. Zu deren Lösung werden so genannte *process mediation patterns* definiert, wie z.B. die Umkehrung zweier Messages. Dadurch kann die Kompatibilität von Web Service Interfaces hergestellt werden, wenn dies a priori nicht gegeben ist.

Wie oben angemerkt, sind Heterogenitäten auf diesen beiden Level sehr häufig anzutreffen. [3] erklärt im Detail, wie und in welchem Umfang diese Mediationstechniken den Nutzbarkeitsgrad von Web Services erhöhen.

5. Frameworks für Semantic Web Services

Nachdem wir die grundlegenden Aspekte erläutert haben, schließen wir mit der kurzen Vorstellung der zwei prominentesten Frameworks für Semantic Web Services. Die meisten Forschungs- und Entwicklungsarbeiten werden im Bezug auf diese umfassenderen Modelle unternommen, und beide wurden als Vorschläge zur Standardisierung beim W3C eingereicht.

5.1. OWL-S

Der chronologisch der erste Ansatz wurde innerhalb des DAML Programms in den USA entwickelt [6]. Unter Verwendung der Web Ontology Language OWL (W3C Recommendation) wird ein Meta-Modell zur semantischen Annotation von Web Services definiert. Dieses besteht aus drei Teilen (siehe Abbildung 5):

1. das **Service Profile** beinhaltet die formale funktionale Beschreibung des Web Service sowie nicht-funktionale Aspekte und dient zum Auffinden und Selektieren
2. das **Service Model** beschreibt die Realisierung des Web Services als einen Prozess. Dies beinhaltet sowohl das Interface zur Konsumierung durch einen Nutzer als auch den Aufruf aggregierter Web Services als Teilprozesse. Die zunächst unzureichende Prozess-Spezifikationssprache wurde später durch eine ausdrucksstarke Prozess Algebra ersetzt.
3. das **Service Grounding** beschreibt die technischen Details zum Aufrufen des Web Service (endpoint, Protokolle, etc.); dies wird durch ein „grounding“ der semantischen Beschreibung zu einer syntaktischen WSDL Beschreibung definiert.

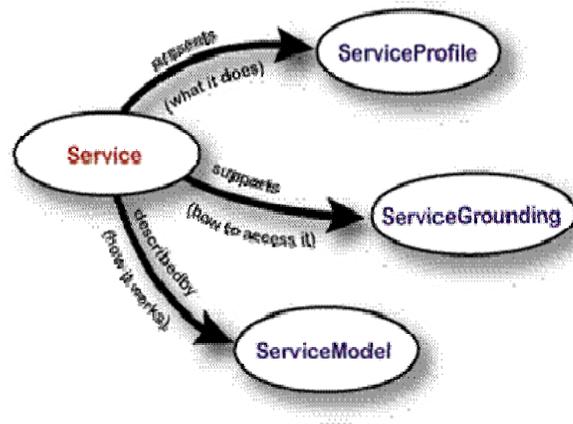


Abbildung 5: OWL-S Meta Model

5.2. Web Service Modeling Ontology (WSMO)

Die Web Service Modeling Ontology WSMO [5] ist der europäische Ansatz zu Semantic Web Services, entwickelt unter der Federführung des Digital Enterprise Research Instituts DERI an der Universität Innsbruck (www.deri.at). Im Gegensatz zu OWL-S definiert WSMO nicht nur ein Modell zur semantischen Beschreibung von Web Services definiert, sondern die semantische Beschreibungen für vier Elemente: **Ontologien** als das Datenmodell, ein Beschreibungsmodell für **Web Services** ähnlich jenem von OWL-S, **Goals** als formal beschriebene Nutzeranfragen, und **Mediators** zum Auflösen und Behandeln von Heterogenitäten (siehe oben).

Dieser werden als die Kernelemente von SOA Systemen verstanden. Dabei sollen End-Nutzer lediglich das zu lösende Problem definieren, während das System automatisch die zu benutzenden Web Services auf Grundlage der semantischen Beschreibungen findet, kombiniert, und ausführt –unter Verwendung von Mediatoren, wie oben erläutert. Damit geht WSMO einen Schritt weiter hin zu *semantic SOA*, also der Realisierung der kompletten SOA Vision auf Grundlage semantischer Technologien. Abbildung 6 zeigt eine Übersicht der WSMO Elemente und ihrem Zusammenhang. Das WSMO Framework bietet dazu eine entsprechende OWL-kompatible Spezifikationsprache [7], sowie eine open source Referenzimplementierung [8].

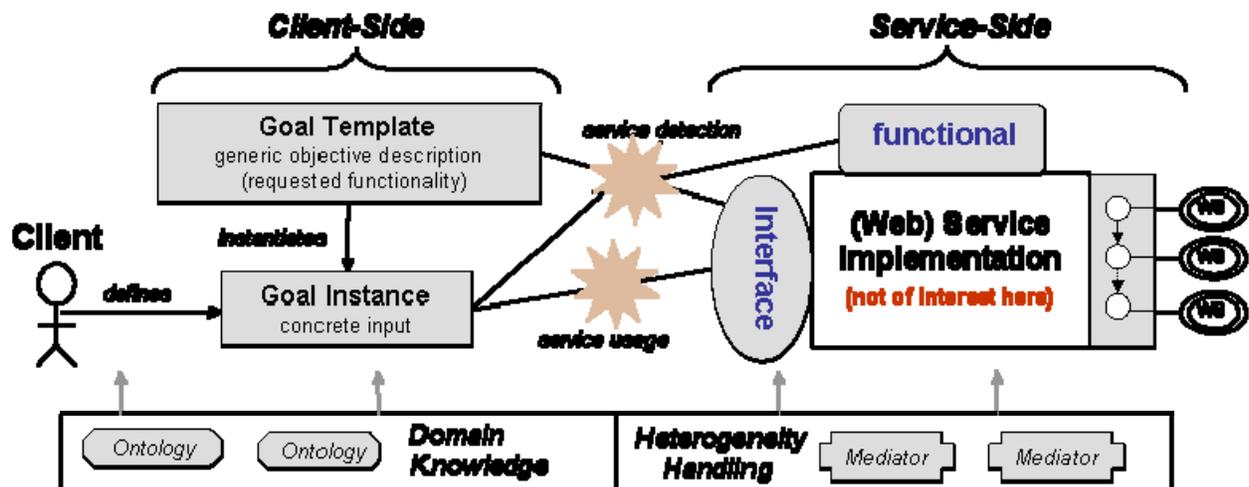


Abbildung 6: Semantic SOA mit WSMO

Referenzen

- [1] Alonso, G. and Casati, F. and Kuno, H. and Machiraju, V.: *Web Services: Concepts, Architectures and Applications*. Springer, 2004.
- [2] Berners-Lee, T.; Hendler, J.; Lassila, O.: The Semantic Web. A new form of Web Content that is meaningful to computers will unleash a revolution of new possibilities. In: *Future of the Web*. *Scientific American*, May 2001.
- [3] Cimpian, E.; Mocan, A.; Stollberg, M.: *Mediation Enabled SemanticWeb Services Usage*. In Proc. of the 1st Asian Semantic Web Conference (ASWC 2006), Beijing, China
- [4] D. Fensel: *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, Berlin, 2001. 2nd Edition. Springer 2003.
- [5] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, and J. Domingue: *Enabling Semantic Web Services: The Web Service Modeling Ontology*. Springer, 2006.
- [6] Martin, D. (ed.): *OWL-S: Semantic Markup for Web Services*. W3C Member Submission 22 November 2004. online: <http://www.w3.org/Submission/OWL-S/>.
- [7] The Web Service Modeling Language WSML. online: www.wsmo.org/wsm1
- [8] The Web Service Execution Environment WSMX, homepage: www.wsmx.org