

Semantic SOA

Überblick und Wertschöpfungspotenzial

Michael Stollberg
Alexander Wahler
Dieter Fensel

STI International
02. Juli 2008

Table of Contents

Abstract (English)	3
1 Einleitung	4
2 Anforderungen und Zielsetzung	5
2.1 Der Web Service Nutzungszyklus.....	5
2.2 Semantische SOA Technologien	7
3 Das WSMO Framework	9
3.1 Zentrale Aspekte	10
3.2 Referenzimplementierung	14
3.3 Andere Ansätze.....	15
4 Anwendungen	16
4.1 Enterprise Application Integration	16
4.2 Business Process Management.....	18
5 Zusammenfassung und Ausblick	20
Referenzen	21
6	21

Abstract (English)

The concept of Service-Oriented Architectures (SOA) proclaims the usage of Web services as the basic building blocks for future IT systems. Although the basic technology stack around WSDL, SOAP, and UDDI supports the technical realization of this vision, it limits the detection of suitable Web services for specific client applications to manual analysis and also does not provide sufficient support for handling expectably occurring incompatibilities. To overcome these deficiencies, several research efforts have developed semantic techniques for the automated discovery, composition, and execution of Web services. One of the most prominent approaches is the Web Service Modeling Ontology WSMO that provides a comprehensive framework for semantically enabled SOA technologies: it takes a goal-based approach and integrates mediation facilities in order to better support and automate the complete Web service usage process within SOA systems. This article explains the essentials of this new technology, discusses its benefits within real-world SOA applications, and gives an outlook to future research and developments.

Keywords: Web Services, SOA, Semantic Technologies, Ontologies, WSMO

1 Einleitung

In Service-Orientierten Architekturen (SOA) sollen Web Services als die Basisbausteine zukünftiger IT Systeme genutzt werden. Die Basistechnologien WSDL, SOAP, UDDI unterstützen die technische Realisierung dieser Vision – zeigen allerdings erhebliche Defizite im Management von Web Services sowie deren Nutzung in SOA Systemen auf. Insbesondere erfordern die rein syntaktischen Beschreibungen eine manuelle Analyse zum Finden und Kombinieren der richtigen Web Services für konkrete Anwendungen. Außerdem können erwartbar auftretende Inkompatibilitäten nur schwer und nicht ausreichend behandelt werden. Die Web Service Modeling Ontology WSMO ist ein prominenter Ansatz zur Lösung dieser Problemstellung durch den Einsatz semantischer Technologien: auf Grundlage eines goal-basierten Ansatzes sowie integrierten Techniken zur Behandlung von Inkompatibilitäten wird ein umfassendes Framework zur Unterstützung und Automatisierung des gesamten Nutzungszyklus von Web Services geboten. Dieser Artikel erläutert die Notwendigkeit und Zielsetzung dieser Technologie, stellt die zentralen Entwicklungen sowie verfügbare open-source Software vor, und veranschaulicht den Nutzen von Semantic SOA Technologien an Hand von Anwendungsbeispielen aus der industriellen Praxis.

Stichworte: Web Services, SOA, Semantische Technologien, Ontologien, WSMO

2 Anforderungen und Zielsetzung

Mit der Einführung von Web Services Ende der 1990er wurde das Konzept von Service-Orientierten Architekturen (SOA) als neues Designparadigma für zukünftige IT Systeme entworfen und seither fortwährend weiterentwickelt. Ein Web Service ist im Prinzip ein gewöhnliches Programm, das durch ein Interface über das World Wide Web aufgerufen werden kann. Die grundlegende Idee von SOA ist, anstatt proprietärer Lösungen Web Services als die Grundbausteine für IT Systeme zu benutzen. Dabei werden allgemein folgende Aspekte als die entscheidenden Vorteile dieser neuen Technologien angesehen:

- die Nutzung des Internet als weltweit verfügbare Infrastruktur für verteilte programmatische Datenverarbeitung
- die Plattformunabhängigkeit der Web Service Technologien, wodurch Anbieter und Nutzer ihre internen Implementierungen beibehalten können
- eine neuartige Technologie zur Behandlung des Integrationsproblems auf intra- und inter-organisatorischer Ebene
- Kostenersparnis für die Entwicklung und Wartung von IT Systemen.

Während die Basistechnologien WSDL, SOAP, und UDDI die grundlegende Bereitstellung von Web Services und deren Verwendung in Client-Applikationen ermöglichen, ist die Entwicklung ausgereifter und benutzerfreundlicher SOA Technologien eine immense Herausforderung. Eine spezifische Problemstellung ist die Unterstützung für das Auffinden nutzbarer Web Services für konkrete Client-Applikationen, was eine zentrale Operation in SOA Systemen darstellt. Dabei erfordern die rein syntaktischen Beschreibungen mit starkem Fokus auf technischen Details eine manuelle Inspektion der einzelnen Web Services, so dass deren angestrebte dynamische Verwendung vor allem in größeren Anwendungen nicht oder nur bedingt realisierbar ist. Ein erfolgversprechender Ansatz zur Lösung dieses Problems ist der Einsatz semantischer Technologien zur besseren Unterstützung der Nutzbarkeitsanalyse von Web Services, die wir im Folgenden genauer vorstellen wollen.

2.1 Der Web Service Nutzungszyklus

Zur Veranschaulichung der Problemstellung inspizieren wir zunächst den Nutzungszyklus von Web Services in SOA Systemen. Abbildung 1 illustriert diesen Prozess, wie er durch die Basistechnologien WSDL, SOAP, und UDDI unterstützt wird; für Einführungen und Details zu diesen Technologien verweisen wir an dieser Stelle auf zahlreiche Internetquellen sowie umfangreiche Fachliteratur (z.B. [1],[4]).

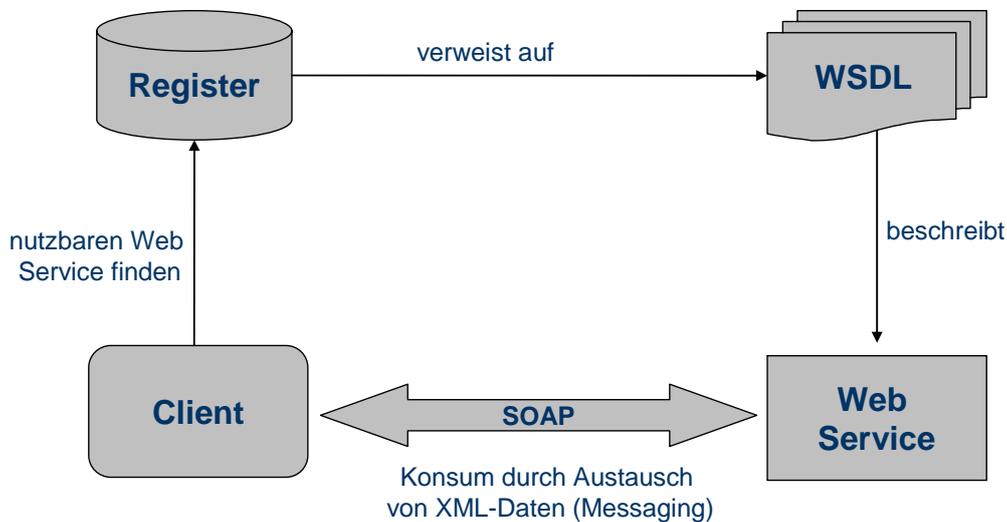


Abbildung 1: Web Service Nutzungszyklus

Der Client – in den meisten Fällen der Entwickler einer Applikation – sucht einen Web Service für eine konkrete Aufgabe, z.B. zum Abrufen einer Auftragsnummer aus dem internen Buchhaltungssystem. Dazu durchsucht er zunächst das *Register*, in dem die verfügbaren Web Services aufgelistet und nach ihrer Funktionalität geordnet sind. Die meisten im Einsatz befindlichen Register-Technologien basieren auf dem UDDI-Modell, also auf zumeist hierarchischen Klassifikationsschemata für die Web Services. Der Client kann daraus mögliche Kandidaten auswählen, deren Verwendbarkeit dann weiter im Detail analysiert werden muss. Dazu muss der Client durch manuelle Inspektion sicherstellen, dass die in der WSDL-Beschreibung definierten Operationen und Datenstrukturen von der Endapplikation unterstützt werden können. Wenn ein nutzbarer Web Service gefunden wurde, kann dieser mit der Hilfe von Client-Stubs in die Endapplikation integriert und schließlich durch den Austausch von SOAP-Messages aufgerufen und konsumiert werden.

Diese technische Unterstützung ist offensichtlich unzureichend, um die Realisierung der SOA Vision in geeigneter Weise zu unterstützen. Dabei hindert vor allem der erhebliche manuelle Analyseaufwand zur Nutzung und Einbindung von Web Services deren umfassende Nutzung als Grundbausteine größerer und vernetzter IT Systeme. Weiterhin können zusätzliche Komplikationen auftreten, die die Nutzung von Web Services in Client-Applikationen behindern – beispielsweise Inkompatibilitäten in den Datenstrukturen oder Änderungen an den vorhandenen Web Services, wie sie in verteilten und heterogenen Systemen erwartbar auftreten. Dies kann durch den Einsatz semantischer Technologien verbessert werden.

2.2 Semantische SOA Technologien

Um die genannten Defizite zu bewältigen, ist das Ziel semantischer SOA Technologien eine integrierte technische Lösung zur besseren Unterstützung und weitgehenden Automatisierung des Web Service Nutzungszyklus. Dazu werden Web Services mit semantischen Beschreibungen versehen, auf deren Grundlage intelligente Mechanismen für die Nutzbarkeitsanalyse sowie zur Behandlung erwartbar auftretender Inkompatibilitäten bereitgestellt werden. Im Folgenden erläutern wir die Grundlagen dieses Ansatzes.

Abbildung 2 zeigt die Struktur von semantischen Beschreibungen für Web Services im Vergleich zu konventionellen WSDL-Beschreibungen. Der erste wichtige Unterschied ist, dass die Terminologie und das nötige Hintergrundwissen in Form von Ontologien beschrieben werden, während in WSDL die benötigten Datenstrukturen in XML-Schemas definiert sind. Eine Ontologie ist eine moderne Wissensrepräsentationstechnik, mit der das konzeptionelle Modell einer Wissensdomäne in Form von Konzepten, Attributen, Relationen, und Axiomen beschrieben wird [5]. Dabei werden vor allem die wesentlichen Eigenschaften und Beziehungen der Entitäten abgebildet, so dass im Gegensatz zu rein technisch motivierten Datenmodellen ein bedeutungsdefinierendes Modell des relevanten Weltausschnitts entsteht. Dieses Modell kann in entsprechenden Ontologiesprachen formalisiert werden, wodurch eine semantische, also bedeutungstragende und -erhaltende Informationsverarbeitung ermöglicht wird [6]. Auf Grund dieser Eigenschaften werden Ontologien als die Basistechnologie für das Semantic Web propagiert, einer Erweiterung des World Wide Web mit semantischen Technologien zur verbesserten Informationsverarbeitung und sowie zur Gewährleistung der Interoperabilität (siehe W3C Semantic Web Activity: www.w3.org/2001/sw/).

Auf dieser Grundlage werden die folgenden Aspekte von Web Services formal beschrieben:

1. die *Funktionalität* des Web Service, gewöhnlich in Form von pre- und postconditions
2. *nicht-funktionale Aspekte* wie Quality-of-Service, Nutzungsrechte, Geschäftsregeln, finanzielle Konditionen, etc. in Form von Regelsystemen
3. das *Interface* des Web Service, wobei zusätzlich zu den Operationen aus der WSDL-Beschreibung der unterstützte Kommunikationsprozess beschrieben wird; optional kann für komplexere Web Services definiert werden, wie dieser andere Web Services zur Realisierung seiner eigenen Funktionalität aggregiert.

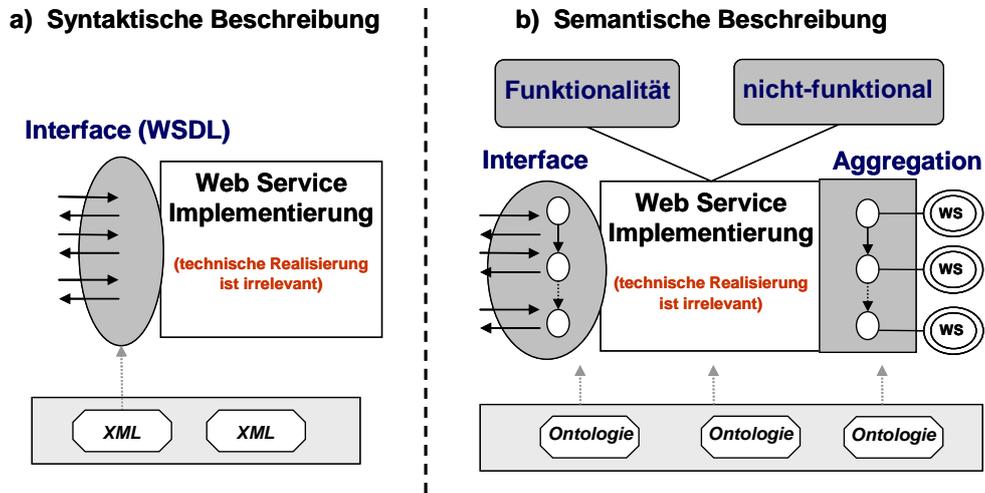


Abbildung 2: Von Syntaktischen zu Semantischen Web Service Beschreibungen

Diese semantischen Beschreibungen beinhalten alle Informationen, die zur Automatisierung der Nutzbarkeitsanalyse von Web Services durch entsprechende Mechanismen nötig sind. Ziel ist es, den Nutzungszyklus von Web Services komplett zu automatisieren, so dass der Client lediglich die Anfrage formuliert und das System automatisch die nötigen Web Services zum Lösen der Anfrage findet und ausführt. Abbildung 3 zeigt diesen Prozess, wobei wir hier die englischen Fachbegriffe beibehalten. In realen Anwendungen wird meist nur ein Teil der Mechanismen verwendet, wie wir in Abschnitt 3 genauer erläutern werden.

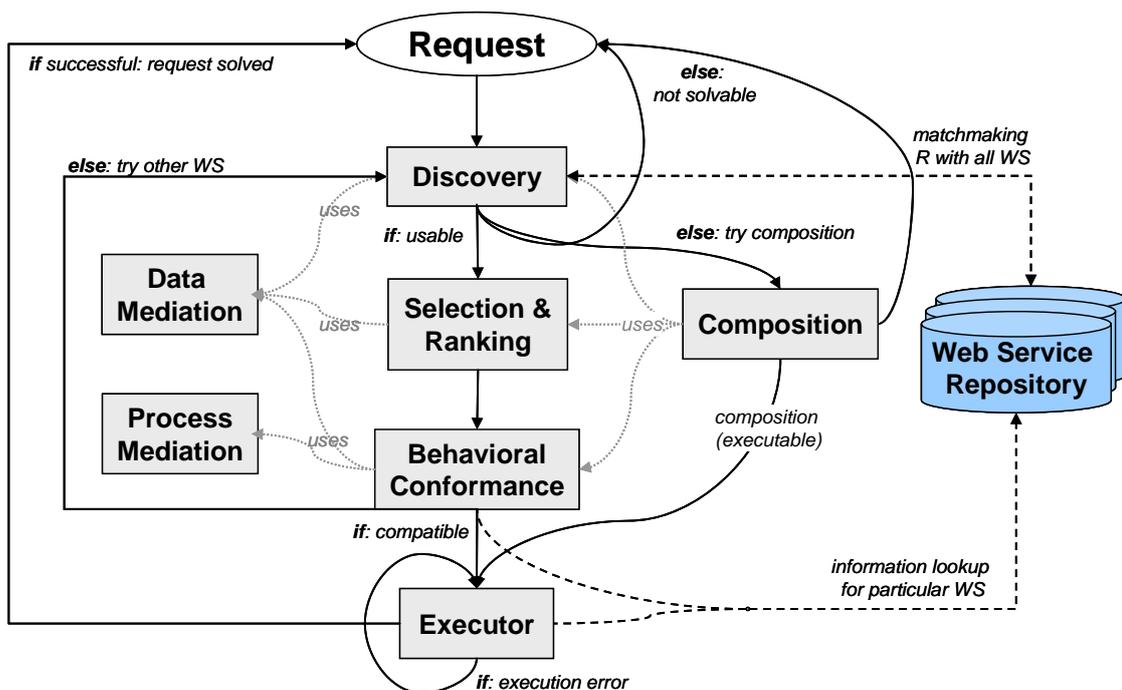


Abbildung 3: Automatisierter Web Service Nutzungszyklus

Zum Lösen einer Client-Anfrage filtert zunächst die *Discovery*-Komponente mögliche Kandidaten aus den vorhandenen Web Services durch semantisches Matchmaking der formalen Funktionalitätsbeschreibungen. Dann werden die Kandidaten in der *Selection & Ranking* - Komponente mit Hinblick auf die nicht-funktionalen Aspekte selektiert, respektive in einer Prioritätsliste geordnet. Als dritter Schritt prüft die *Behavioral Conformance* - Komponente, ob der Client erfolgreich mit den nötigen Web Service kommunizieren kann. Damit ist die Nutzbarkeitsanalyse abgeschlossen, und die gefundenen Web Services werden automatisch ausgeführt, um die Anfrage zu lösen. Wenn kein direkt nutzbarer Web Service vorhanden ist, wird in der *Composition*-Komponente mit Planungstechniken versucht, eine ausführbare Kombination mehrerer Web Services zum Lösen der Anfrage zusammenzustellen. Falls erforderlich, können die Komponenten semantische Techniken zum Behandeln auftretender Inkompatibilitäten nutzen, insbesondere für inkompatible Daten- und Ontologiedefinition (*Data Mediation*) sowie zur Sicherstellung eines kompatiblen Kommunikationsprozesses zwischen dem Client und den Web Services (*Process Mediation*).

Zur Sicherstellung einer hohen Qualität für die Nutzbarkeitsanalyse verwenden die einzelnen Mechanismen inferenz-basierte Techniken, wobei entsprechende Entwicklungen aus der Künstlichen Intelligenz (KI) als Basis dienen. In den letzten Jahren ist eine Vielzahl von Forschungsarbeiten auf diesem Gebiet entstanden, und inzwischen werden semantische SOA Technologien auch in den wichtigen internationalen Standardisierungsgremien behandelt (W3C, OASIS, OMG). Im Folgenden stellen wir einen konkreten Ansatz für die Realisierung semantischer SOA Technologien vor.

3 Das WSMO Framework

Die Web Service Modeling Ontology WSMO ist ein umfassendes Framework für semantische SOA Technologien. Im Gegensatz zu anderen Ansätzen beinhaltet es neben der Verwendung von Ontologien zur semantischen Beschreibung von Web Services zwei weitere Elemente, die zur ganzheitlichen Unterstützung des Web Service Nutzungszyklus erforderlich erscheinen: (1) einen goal-basierter Ansatz zur besseren Unterstützung von Clients bei der Verwendung von Web Services, und (2) integrierte Techniken zur Behandlung von erwartbar auftretenden Inkompatibilitäten. Des Weiteren wird eine Referenzimplementierung geboten, die als open-source Projekt entwickelt wird und frei verwendbar ist.

Das WSMO Framework wird seit 2004 durch ein internationales Konsortium aus hauptsächlich europäischen Forschungs- und Industriepartnern entwickelt, und ist in zahlreichen Forschungsprojekten erfolgreich eingesetzt und weiterentwickelt worden. Im Folgenden erläutern wir die zentralen Aspekte; für weiterführende Informationen verweisen wir auf ein detailliertes Buch [5] sowie die WSMO Homepage (siehe: www.wsmo.org).

3.1 Zentrale Aspekte

Das WSMO Framework besteht aus vier Kernelementen (siehe Abbildung 4): (1) *Ontologien* definieren die Terminologie und das nötige Hintergrundwissen für die (2) semantischen Beschreibungen von *Web Services*, (3) *Goals* beschreiben Client-Anfragen und ermöglichen damit die problemorientierte und dynamische Nutzung von *Web Services*, und (4) *Mediatoren* stellen Techniken zum Behandeln von Inkompatibilitäten zur Verfügung. Wir erläutern nun, warum diese Elemente zur Realisierung semantischer SOA Technologien notwendig sind, und wie sie in WSMO definiert und gehandhabt werden.

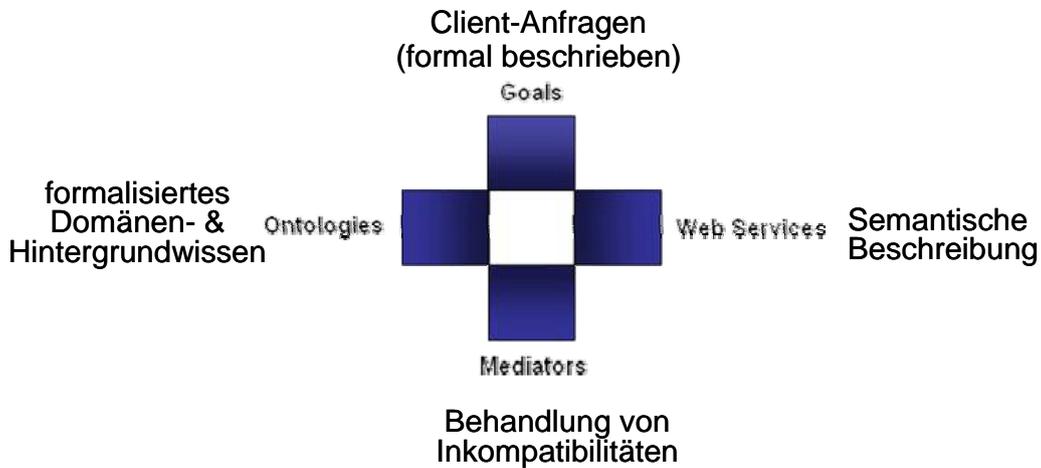


Abbildung 4: WSMO Top-Level Elemente

Ontologien und Semantische Web Service Beschreibung

Zunächst illustrieren wir die semantische Beschreibung von *Web Services* auf der Basis von *Ontologien*. Dafür bietet WSMO mit der *Web Service Modeling Language* *WSML* eine eigene Spezifikationsprache, die neben der Unterstützung für konzeptionellen Modellierung aller WSMO Elemente eine formale *Ontologiesprache* mit verschiedenen logischen Varianten darstellt [2]. Abbildung 5 illustriert die Definition einer *Ontologie* sowie die Beschreibung eines *Web Service* in *WSML*.

Als Beispiel dient hier ein einfacher *Web Service*, der für eine gegebene Telefonnummer den Standort des Anschlusses ermittelt. Die *Ontologie* auf der linken Seite definiert das benötigte Hintergrundwissen: eine Telefonnummer ist in Landes- und Ortsvorwahl sowie die Anschlussnummer untergliedert. Die Zuordnung von Vorwahlen zu Ländern und Städten ist in einer *Knowledge Base* in Form von Instanzen definiert. Die Länder und deren Städte sind in der externen *Location-Ontologie* definiert, die über den Namespace „loc“ referenziert wird.

```

ontology _"http://www.wsmo.org/ontologies/telephony.wsmi"
namespace { _"http://www.wsmo.org/ontologies/telephony#",
  loc _"http://www.wsmo.org/ontologies/location#" }
concept phoneNumber
  countryCode ofType countryCode
  areaCode ofType areaCode
  number ofType _string
instance STlofficeNumber memberOf phoneNumber
  countryCode hasValue "+43"
  areaCode hasValue "0512"
  number hasValue "5076488"
concept TelephoneCode
  code ofType _string
concept countryCode subConceptOf TelephoneCode
  country ofType loc#country
concept areaCode subConceptOf TelephoneCode
  area ofType loc#city
instance AustriaCode memberOf countryCode
  code hasValue "+43"
  country hasValue loc#austria
instance InnsbruckCode memberOf areaCode
  code hasValue "0512"
  city hasValue loc#innsbruck

webService _"http://www.wsmo.org/services/phoneLocation.wsmi"
namespace { _"http://www.wsmo.org/services/phoneLocation#",
  dc _"http://purl.org/dc/elements/1.1#",
  tel _"http://www.wsmo.org/ontologies/telephony#",
  loc _"http://www.wsmo.org/ontologies/location#" }
annotation
  dc#title hasValue "Lookup Service: Location by Phone Number"
  dc#description hasValue "provides the city of a given phone number"
importsOntology {
  _"http://www.wsmo.org/ontologies/telephony.wsmi",
  _"http://www.wsmo.org/ontologies/location.wsmi"}
capability
  sharedVariables {?number}
  precondition
  definedBy
    ?number memberOf tel#phoneNumber.
  postcondition
  definedBy
    ?location memberOf loc#city and
    ?number[areaCode hasValue ?location].

```

Abbildung 5: Semantische Beschreibung eines Web Services in WSMI

Die Web Service Beschreibung auf der rechten Seite beginnt mit dem Identifier, der als IRI (Internationalized Resource Identifier) definiert ist. Danach werden Namespaces definiert, um die verwandten Ontologien zu referenzieren. Der „annotations“-Block enthält eine natürlich-sprachliche Beschreibung des Web Services, wozu der Dublin Core Metadata Standard verwendet wird (siehe <http://dublincore.org/>). Es folgt die Angabe der verwandten Ontologien, und danach folgt die formale Funktionsbeschreibung des Web Service. Diese heißt „capability“ in WSMO, und wird durch pre- und postconditions in Form logischer Axiome definiert. Hier wird als Input eine Telefonnummer verlangt, und der Web Service gibt die zugehörige Stadt zurück; durch die „shared variables“ kann der logische Zusammenhang der Vor- und Nachbedingungen explizit definiert werden.

Damit wird präzise beschrieben, was der Web Service macht. Auf dieser Grundlage kann mit entsprechenden matchmaking-Techniken sehr genau festgestellt werden, ob der Web Service zum Lösen einer Client-Anfrage geeignet ist oder nicht. Dabei stellen die Ontologien die eindeutige semantische Definition der relevanten Datenstrukturen sicher, wodurch außerdem mögliche Missverständnisse verhindert werden können.

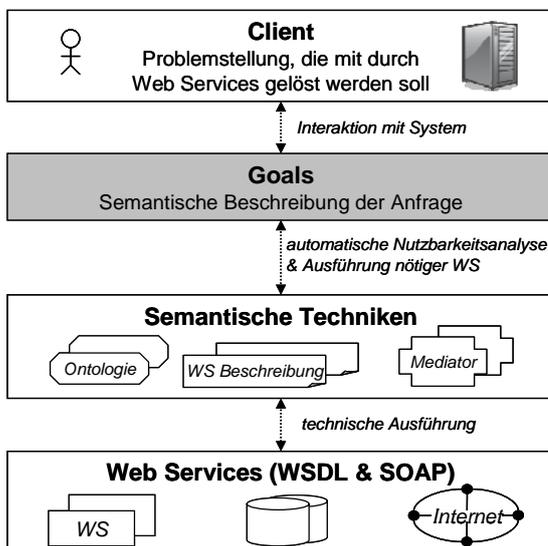
Goals – Problemorientierte und Dynamische Nutzung von Web Services

Ein wichtiger Aspekt für funktionstüchtige SOA Technologien ist die Unterstützung von Clients in der Suche, Inspektion, und Einbindung von Web Services in eine Applikation. Während dies in den meisten Systemen vernachlässigt wird, bietet WSMO einen goal-basierten Ansatz zur automatischen und benutzerfreundlichen Verwendung von Web Services.

Ein Goal ist die semantische Beschreibung des Problems, das der Client durch die Nutzung von Web Services lösen möchte, wobei von technischen Details abstrahiert wird. Der Client formuliert Anfragen als Goal, und das System findet automatisch die nötigen Web Services und führt diese aus. Damit werden zwei

zentrale Verbesserungen im Vergleich zu bestehenden SOA Technologien geschaffen (siehe Abbildung 6). Erstens wird problemorientierte Nutzung von Web Services ermöglicht: die Client-System Interaktion findet auf der Ebene von Goals statt, so dass Clients sich nicht mehr mit technischen Details beschäftigen müssen, sondern sich auf das zu lösende Problem konzentrieren können. Zweitens kann die Flexibilität des Systems durch die dynamische Auswahl von einem der nutzbaren Web Services zur Laufzeit verbessert werden. Damit wird die Problematik der statischen Einbindung von Web Services durch WSDL-Stubs in Client-Applikationen behoben: dabei ist nämlich die Client-Applikationen nicht mehr funktionsfähig, wenn der betreffende Web Service nicht verfügbar ist oder geändert wird.

(1) Problemorientierte Nutzung von Web Services



(2) Dynamische Nutzung von Web Services

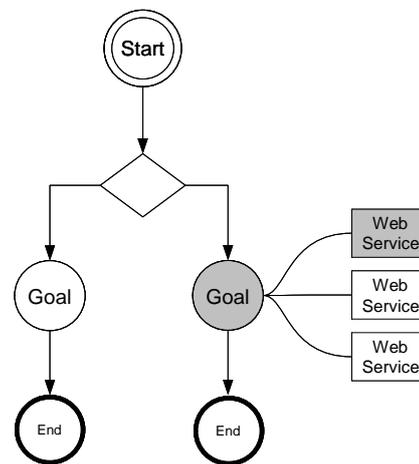


Abbildung 6: Der Goal-basierte Ansatz zur Unterstützung von Clients in SOA Systemen

Zur technischen Umsetzung dieses Ansatzes werden in WSMO *Goal Templates* und *Goal Instances* unterschieden. Ein Goal Template ist eine generische Problembeschreibung, die im System gespeichert wird. Goal Instances beschreiben konkrete Client-Anfragen, und werden durch die Instantiierung eines geeigneten Goal Templates mit konkreten Input-Daten definiert. Ein Beispiel: für eine gegebene Telefonnummer oder Postleitzahl soll der physikalische Standort gefunden werden. Das Goal Template beschreibt dies durch eine *requested capability* analog zu der Web Service Beschreibung aus Abbildung 5 wie folgt:

```

precondition    ?x memberOf phoneNumber or ?x memberOf ZIPcode.
postcondition   ?location memberOf city and (
                    (?x[areaCode hasValue ?location] memberOf phoneNumber) or
                    (?x[belongsToCity hasValue ?location] memberOf ZIPcode) ).
    
```

Für eine konkrete Anfrage instantiiert der Client die Input-Variable $?x$ mit Daten – z.B. mit der Instanz `STIofficeNumber` aus der Ontologie in Abbildung 5. Dann ist der oben beschriebene Web Service nutzbar zum Lösen der Goal Instance; dies ist nicht gegeben, wenn $?x$ mit einer Postleitzahl instantiiert wird. Diese Konzipierung ermöglicht effiziente, skalierbare, und nutzerfreundliche semantische SOA Technologien. Zunächst kann der Nutzungszyklus in Design- und Laufzeitoperationen aufgeteilt werden: zur Designzeit werden die erwartbar aufwendigen und kostenintensiven Operationen für die automatisierte Nutzbarkeitsanalyse auf der Ebene von Goal Templates durchgeführt; zur Laufzeit muss dann lediglich eine Auswahl aus den nutzbaren Web Services getroffen werden. Des Weiteren wird die Formulierung von Goals durch Clients erheblich vereinfacht (siehe Abbildung 7).

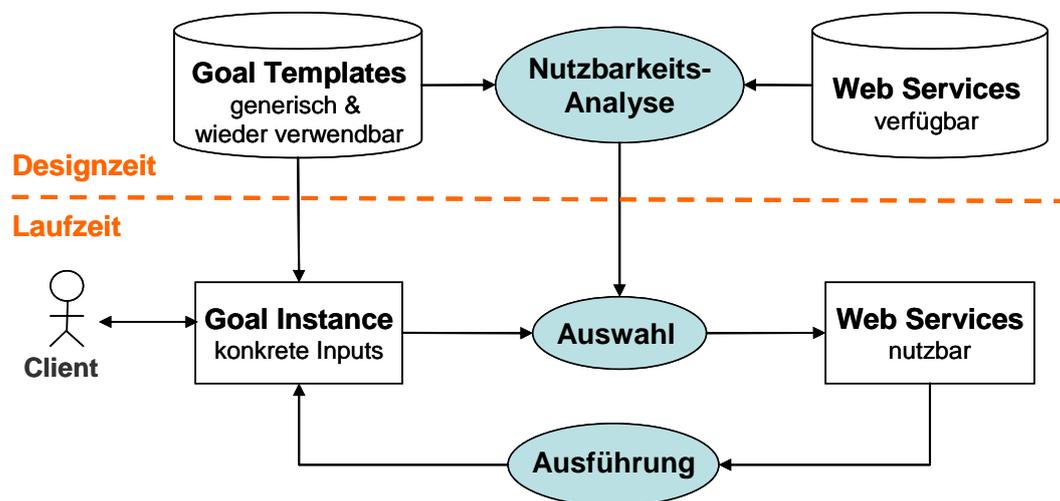


Abbildung 7: Goals

Mediatoren – Behandlung von Inkompatibilitäten

Das vierte Element für die umfassende Realisierung semantischer SOA Technologien ist die Integration von Mediations-Techniken zur Behandlung von Inkompatibilitäten. Diese können auf verschiedenen Ebenen auftreten und die erfolgreiche Interaktion zwischen Client und Web Service behindern – z.B. durch verschiedene Spezifikations Sprachen, unterschiedliche Datenstrukturen, oder inkompatible Kommunikationsprozesse. Allgemein bekannt ist dies als das Integrationsproblem, in das heutzutage ein großer Anteil der IT Investitionen fließen [3].

Ein wesentlicher Vorteil von semantischen Technologien ist, dass generische Techniken zur Behandlung von Inkompatibilitäten bereitgestellt werden können. Als eine der wichtigsten Inkompatibilitätsprobleme können heterogene Datenstrukturen durch das Mapping von Ontologien integriert werden, wobei die Mappings auf Grundlage der eindeutig definierten Meta-Struktur von Ontologien definiert sind. Als Beispiel hierzu illustriert Abbildung 8 die Integration von Daten aus zwei verschiedenen Ontologien, die Personen mit unterschiedlichen konzeptionellen Modellen beschreiben: Ontologie A unterscheidet Erwachsene

und Kinder als Sub-Konzepte, während Ontologie B lediglich das Konzept *Person* mit dem Attributen *Name* und *Alter* definiert. Das Mapping unten in der Abbildung definiert, dass eine Instanz von *Person* aus Ontologie B eine Instanz von *Adult* aus Ontologie A ist, wenn diese 18 Jahre oder älter ist. Damit kann jede Instanz von Ontologie B in eine Instanz von Ontologie A transformiert werden, so dass der Datenaustausch unter Erhalt der jeweiligen Semantik der ontologischen Modelle ermöglicht wird. Auf dieser Grundlage stellt das WSMO Framework umfangreiche Techniken zur Behandlung von Inkompatibilitäten bereit [8].

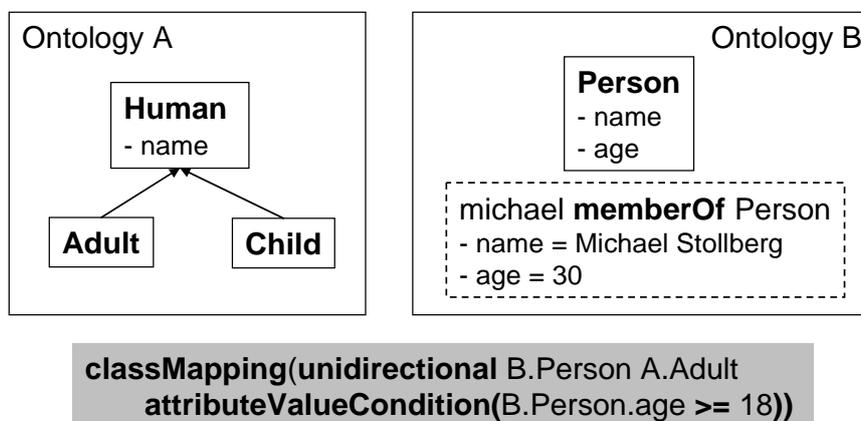


Abbildung 8: Beispiel zur ontologie-basierten Datenintegration

3.2 Referenzimplementierung

Das Web Service Execution Environment WSMX stellt die Referenzimplementierung des WSMO Frameworks dar. Darin sind alle Mechanismen für die Automatisierung des Web Service Nutzungszyklus implementiert, auf Basis des goal-basierten Ansatzes sowie mit integrierten Mediations-Techniken zur Behandlung von Inkompatibilitäten wie oben erläutert.

WSMX ist als Middleware-Lösung für peer-to-peer SOA-Systeme konzipiert. Web Services werden von den Anbietern im System registriert und mit semantischen Beschreibungen versehen; die Clients formulieren ihre Anfragen als Goals, und WSMX findet die nötigen Web Services und initiiert den Datenaustausch über das Web. Abbildung 9 zeigt eine Übersicht über das WSMX System. Das oberste Layer stellt mit dem Web Service Modeling Toolkit WSMT eine umfangreiche Entwicklungsplattform bereit [7]. Das Broker-Layer beinhaltet die Mechanismen zur automatischen Nutzbarkeitsanalyse wie oben beschrieben (siehe Abbildung 3), und das Base-Layer stellt die nötigen Basistechniken zur Verfügung.

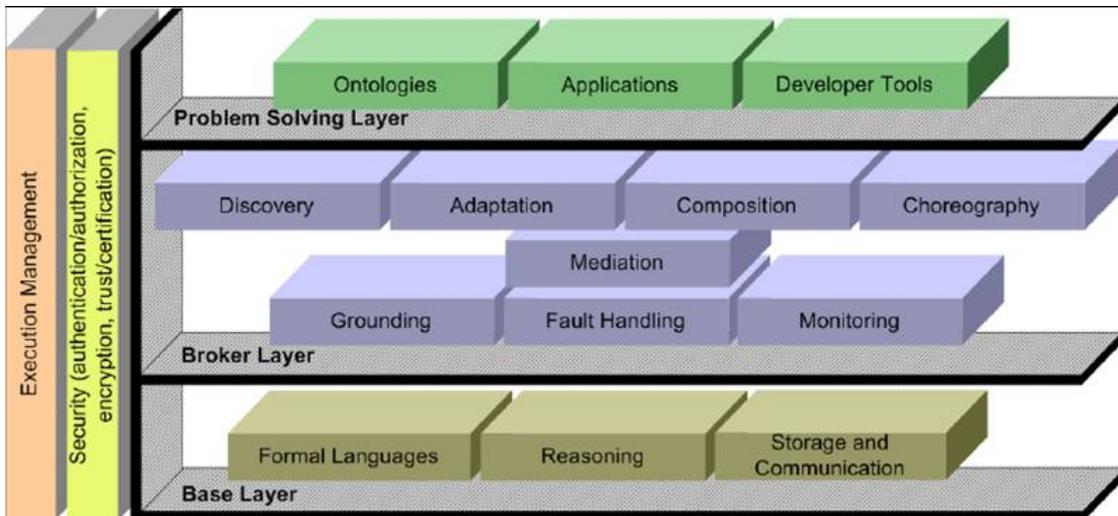


Abbildung 9: Überblick Web Service Execution Environment

WSMX ist als open-source System implementiert, das unter der LGPL Lizenz (GNU Lesser General Public License) zur freien Verfügung steht (siehe Homepage: www.wsmx.org). Neben umfangreicher Dokumentation umfasst die Software auch:

- API zum programmatischen Management von WSMO-Elementen
- Validierungsservice für WSMO-Definition
- Reasoning-Services für die WSML-Spezifikationssprache.

3.3 Andere Ansätze

Das Interesse an semantische SOA Technologien hat in den letzten Jahren enorm zugenommen. Neben WSMO haben insbesondere die folgenden beiden Ansätze Aufmerksamkeit in der Forschung genossen:

(1) OWL-S – Semantische Beschreibung von Web Services

Das OWL-S Modell wurde zwischen 2002 und 2006 als erster Ansatz zur semantischen Beschreibung von Web Services im Zuge des DAML-Programms in den USA entwickelt (siehe www.daml.org/services/owl-s/). Darin wurde die grundlegende Beschreibungsstruktur definiert (siehe Abbildung 2). Im Gegensatz zu WSMO werden aber weder die Unterstützung von Clients noch die Behandlung von Inkompatibilitäten abgedeckt.

(2) SAWSDL – Semantische Annotation von WSDL

Im August 2007 wurde SAWSDL als erste W3C-Recommendation für semantische Web Service Technologien veröffentlicht (siehe Homepage: www.w3.org/2002/ws/sawSDL/). Dies unterstützt die semantische Annotation von WSDL-Beschreibungen durch zusätzliche Tags, mit denen Referenzen zu Ontologien definiert werden – z.B. dass ein XML-Datentyp sich auf ein

bestimmtes Konzept bezieht. Obwohl derartige semantische Beschreibungen lediglich marginale Verbesserungen zur Nutzbarkeitsanalyse erlauben, sind auf Grund der Veröffentlichung durch das W3C weitere Entwicklungen zu erwarten.

4 Anwendungen

Nachdem wir den Bedarf an semantischen SOA Technologien sowie Frameworks für die technische Realisierung erläutert haben, wollen wir nun an Hand praktischer Beispiele deren Verwendung und insbesondere den wirtschaftlichen Mehrwert illustrieren.

4.1 Enterprise Application Integration

Das erste Beispiel beschäftigt sich mit der Daten- und Serviceintegration im B2B-Bereich. Dazu stellen wir eine mit WSMO-Technologien erweiterte Großhandelsplattform für den Telekommunikationssektor vor, die im Rahmen des EU-Forschungsprojekts DIP (2004 – 2006, Homepage: dip.semanticweb.org) für die British Telekom (BT) entwickelt wurde.

BT Wholesale ist eine klassische Supply-Chain-Lösung zur Erfassung und Verwaltung der Produkte und Dienstleistungen verschiedener Telekom-Zulieferer, die dann für die in- und externe Nutzung bereitgestellt werden. Das System umfasst ca. 150 Lieferanten, die vor allem Basisdienste wie Hardware, Netzwerklösungen, Internet-Pauschalpakete, etc. anbieten. Die Aufgabe von *BT Wholesale* die Beschaffung und Bereitstellung sowie die Qualitätskontrolle und die Abrechnung der Zuliefererdienste. Das System ist weitgehend automatisiert: die Zulieferer bieten ihre Dienste über technische Schnittstellen an (zunehmend in Form von Web Services), die dann in *BT Wholesale* verwaltet und über entsprechende Schnittstellen den Klienten zugänglich gemacht werden.

Die technische Umsetzung und Wartung des Systems ist sehr aufwendig. Die Einbindung eines neuen Zulieferers nimmt durchschnittlich 4 Monate in Anspruch, wobei die Integration inkompatibler Daten- und Informationsstrukturen sowie der Aufbau der nötigen technischen Infrastruktur den Großteil ausmacht. Die einzelnen Zulieferer nutzen oft unterschiedliche Technologien für das interne Management, und haben meist wenig Erfahrung mit gängigen B2B-Integrationstechnologien. Daher muss *BT Wholesale* erhebliche Ressourcen zur Wartung des Systems aufbringen. Die Kosten dafür betragen ca. € 1,5 Million jährlich.

Diese Situation soll durch den Einsatz der WSMO-Technologie verbessert werden. Dazu wurde zunächst eine zentrale Ontologie erstellt, in der die relevanten Produktarten sowie das nötige Wissen für die Qualitätskontrolle und das Abrechnungssystem definiert sind. Die Ontologien basiert auf entsprechenden Standards aus dem Telekomsektor, um die Interoperabilität mit Systemen anderer Anbieter zu gewährleisten. Auf dieser Basis wurden die Angebote und Dienste der Zulieferer semantisch beschrieben, so dass diese als *WSMO Web Services* behandelt werden können. Das *BT Wholesale* System nutzt WSMX zur Verwaltung der vorhandenen Angebote, und die Endnutzer

können Anfragen und Aufträge in Form von *WSMO Goals* formulieren. Abbildung 10 zeigt die Struktur des Systems.

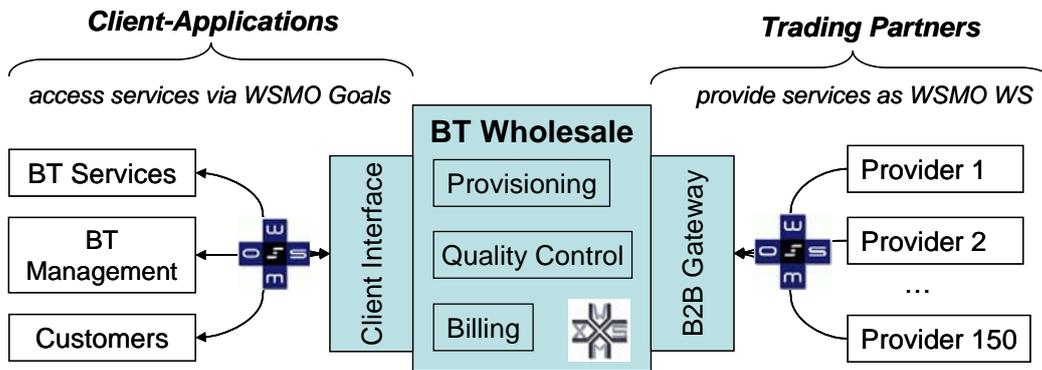


Abbildung 10: WSMO-basierte Telekom-Großhandelsplattform

Erfahrungstests mit diesem System zeigen entscheidende Vorteile für die Integration von Zulieferern und die Verwaltung der Angebote im *BT Wholesale* System. Zulieferer müssen lediglich ihre Dienste als Web Service anbieten und mit einer entsprechenden semantischen Beschreibung versehen. Diese sind relativ einfach zu erstellen: da die meisten Produkte und Dienstleistungen strukturell sehr ähnlich sind und sich lediglich in Details unterscheiden, kann die Erstellung der nötigen WSMO-Beschreibungen über Templates auf Basis der zentralen Ontologie unterstützt werden. Weiterhin können auftretende Inkompatibilitäten mit den Mediationstechniken explizit und bedeutungserhaltend aufgelöst werden. Eine abschließende Erfolgsstudie zeigt, dass damit der Integrationsaufwand für neue Zulieferer auf ca. 2 Monate verkürzt werden kann, wodurch eine jährliche Kostenersparnis von € 800.000,- für *BT Wholesale* erreicht werden kann.

Neben der Kostensenkung kann außerdem die funktionale Qualität des Systems verbessert werden. Auf Grund der semantischen Beschreibung kann eine sehr viel genauere Klassifikation der vorhandenen Angebote vorgenommen werden, z.B. durch den Einsatz automatischer Discovery-Techniken. Außerdem kann mit dem goal-basierten Ansatz die Unterstützung für Endnutzer erheblich verbessert werden: während zuvor lediglich manuelle Suchtechniken vorhanden waren, können in- und externe Nutzer konkrete Anfragen als Goals formulieren und automatisch bearbeiten lassen. Ein weiterer Vorteil ist, dass die semantische Beschreibung von Angeboten und Informationen explizit definiert ist, während dies zuvor in Dokumenten und Kalkulationstabellen versteckt war. Zusammenfassend lassen sich folgende Vorteile semantischer SOA Technologien für *BT Wholesale* festhalten:

- Senkung der Integrationskosten um 40 – 60 %
- Funktionale Qualitätsverbesserungen durch semantische Informationsverarbeitung
- Gewährleistung der Interoperabilität durch standardisierte Ontologien und ontologie-basierte Datenintegration

- Basis für weitere neuartige und innovative Anwendungen, z.B. die dynamische Bündelung von Produkten mit Hilfe automatisierter Kompositionstechniken.

4.2 Business Process Management

Als zweites Anwendungsfeld erläutern wir die Verwendung semantischer Technologien zur Verbesserung von Business Process Management Technologien (BPM). Dies beschäftigt sich mit der technischen Unterstützung für die Definition, Ausführung, und Kontrolle von Geschäftsprozessen innerhalb sowie zwischen Organisationen. Ein Geschäftsprozess wird gewöhnlich in zwei Perspektiven unterteilt: die obere Managementebene beschäftigt sich mit der Spezifikation des Gesamtprozesses mit Hinblick auf die wirtschaftlichen Anforderungen, und auf der unteren Implementierungsebene werden die nötigen informationstechnischen Lösungen zur Ausführung des Prozesses bereitgestellt. In den letzten Jahren wurde eine Vielzahl neuer BPM-Technologien entwickelt, die unter anderem Web Services und das SOA-Prinzip als Basistechnologien nutzen.

Der zentrale Standard dafür ist die Business Process Execution Language (BPEL), eine Spezifikations- und Ausführungssprache für Geschäftsprozesse mit Unterstützung für die Einbindung und Nutzung von Web Services auf der Implementierungsebene. Ein BPEL-Prozess besteht auf der Managementebene aus einer gewöhnlichen Prozessstruktur. Auf der Implementierungsebene werden die einzelnen Aktivitäten durch Web Services realisiert, um die genannten Vorteile dieser neuen Technologie zu nutzen. Dabei werden die Web Services über konventionelle SOA Technologien eingebunden, also via WSDL-Stubs direkt aufgerufen. Somit können Geschäftsprozesse aus Web Services zusammengesetzt und über das Internet ausgeführt werden – allerdings treten dabei die gleichen Problemstellungen auf wie eingangs diskutiert. Der Entwickler muss die brauchbaren Web Services manuell suchen und einbinden sowie eventuell auftretende Inkompatibilitäten behandeln, so dass die technische Umsetzung eines Geschäftsprozesses einen hohen manuellen Integrationsaufwand erfordert. Außerdem ist die Flexibilität und Stabilität durch die statische Einbindung der Web Services stark eingeschränkt: wenn einer der genutzten Web Services nicht verfügbar ist oder geändert wird, dann ist der gesamte Geschäftsprozess nicht mehr ausführbar.

Diese Defizite können durch die Verwendung semantischer SOA Technologien behoben werden. Eine spezifische Lösung dafür ist *Semantic BPEL* – eine die Erweiterung des BPEL-Standards mit der WSMO-Technologie, die derzeit in dem EU-Forschungsprojekt SUPER entwickelt wird (Homepage: www.ip-super.org). Abbildung 11 zeigt die grundlegende Struktur des Ansatzes: anstatt der statischen Einbindung von Web Services werden die einzelnen Aktivitäten eines Geschäftsprozesses als *WSMO Goals* definiert. Zur Laufzeit werden dann die konkreten Web Services mit Hilfe des WSMX-Systems automatisch ausgewählt und ausgeführt, wobei die oben erläuterte Trennung von Goal Templates und Goal Instances die Effizienz und Skalierbarkeit gewährleistet (siehe Abbildung 7). Das nötige Hintergrundwissen ist in Ontologien definiert, auf deren Grundlage die Prozesse, Goals, und Web Services semantisch beschrieben werden. Damit

werden die beiden genannten zentralen Probleme gelöst: (1) die Nutzbarkeitsanalyse und Einbindung von Web Services in BPEL-Prozesse kann automatisiert werden, und (2) die Flexibilität und Stabilität kann durch die dynamische Auswahl der konkreten Web Services zur Laufzeit signifikant verbessert werden.

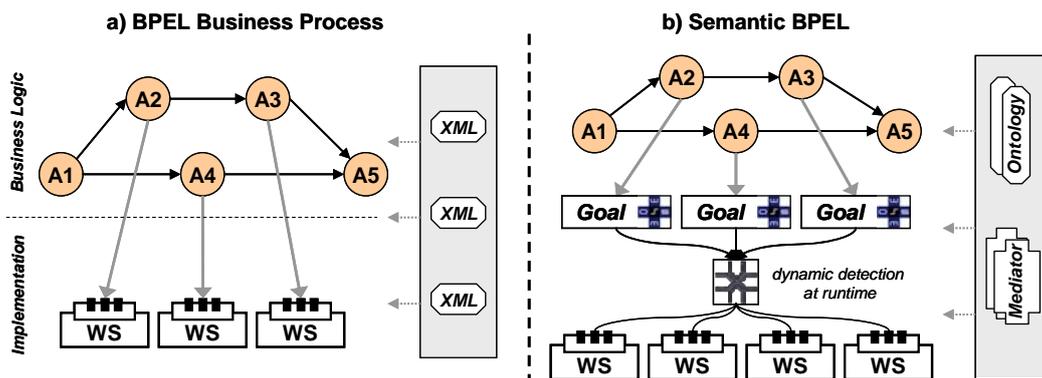


Abbildung 11: WSMO-basierte Erweiterung von BPEL

Im Rahmen des SUPER-Projekts wird eine Entwicklungs- und Ausführungsplattform für Semantic BPEL entwickelt und in verschiedenen Anwendungen eingesetzt und getestet. Dies sind vornehmlich umfangreiche Verwaltungs- und Steuerungssysteme größerer Telekom-Anbieter, in deren Geschäftsprozessen mehrere hundert Web Services aus verschiedenen Unternehmensbereichen integriert werden. Dabei erfordert die hohe Marktdynamik eine flexible Anpassung der Geschäftsprozesse zur Sicherung der Wettbewerbsfähigkeit, und es werden kontinuierlich neue Produkte und Dienste entwickelt, die in das System einzubinden sind. Dafür lassen sich aus den ersten Testergebnissen folgende Vorteile von Semantic BPEL ersehen, die jenen aus dem ersten Anwendungsbeispiel ähnlich sind:

- erhebliche Senkung der Entwicklungs- und Wartungskosten durch Automatisierung der Nutzbarkeitsanalyse von Web Services für deren Einbindung in Geschäftsprozesse
- Verbesserung der Flexibilität und Stabilität des Systems durch die dynamische Auswahl der konkreten Web Services zur Laufzeit
- Funktionale Qualitätsverbesserungen durch semantische Informationsverarbeitung
- Verbesserung der menschlichen Kommunikation zwischen Management- und Implementierungsebene durch die semantische und explizite Beschreibungen der relevanten Terminologie und Ressourcen (bzgl. Business-IT-Divide).

5 Zusammenfassung und Ausblick

Dieser Artikel hat einen vertiefenden Einblick in semantische SOA Technologien gegeben. Ziel ist, den Nutzungszyklus von Web Services zu automatisieren und die Interoperabilität in SOA-Systemen zu gewährleisten. Dies wird durch die Basistechnologien WSDL, SOAP, und UDDI nur unzureichend unterstützt, da die Auswahl und Integration nutzbarer Web Services in eine Client-Applikation erheblichen manuellen Analyseaufwand erfordert. Um dies zu verbessern, werden automatisierte Mechanismen durch Unterstützung der gesamten Nutzbarkeitsanalyse von Web Services sowie Techniken zur Behandlung erwartbar auftretender Inkompatibilitäten zur Verfügung gestellt. Diese arbeiten auf der Grundlage semantischer Beschreibungen, wobei die Terminologie und das nötigen Hintergrundwissen in Ontologien als bedeutungstragende Wissensrepräsentation definiert sind.

Wir haben die Web Service Modeling Ontology WSMO als ein umfassendes Framework zur Realisierung semantische SOA Technologien vorgestellt. Darin ermöglicht der goal-basierte Ansatz die problemorientierte und dynamische Nutzung von Web Services, und die integrierten Techniken unterstützen die Behandlung aller relevanten Inkompatibilitäten. Die vorgestellten Anwendungsszenarien aus der industriellen Praxis zeigen, dass neben funktionalen Verbesserungen auch wirtschaftliche Vorteile durch die signifikante Reduktion der Entwicklungs- und Wartungskosten vor allem für größere SOA-Systeme erreicht werden können.

Dies verdeutlicht, dass semantische Technologien sinnvoll und geeignet erscheinen, um die SOA-Vision besser in die Realität umzusetzen. Die internationale Standardisierung auf diesem Gebiet hat bereits begonnen, und die hier vorgestellten Technologien sind schon erfolgreich in industriellen Applikationen eingesetzt worden. Für die Weiterentwicklung semantischer SOA-Technologien beschäftigen sich derzeit beginnende Forschungsprojekte von allem mit der Skalierbarkeit und Effizienz sowie mit der umfassenden Unterstützung zur Entwicklung von Ontologien und der semantischer Beschreibungen.

Referenzen

- [1] Gustavo Alonso, Fabio Casati, Harumi Kuno, H., Vijay Machiraju: *Web Services: Concepts, Architectures, and Applications*. Data-Centric Systems and Applications. Springer, Berlin, Heidelberg, 2004.
- [2] Jos de Bruijn, Dieter Fensel, Mick Kerrigan, Uwe Keller, Holger Lausen, James Scicluna: *Modeling Semantic Web Services: The Web Service Modeling Language*. Springer, 2008.
- [3] Christoph Bussler: *B2B Integration*. Springer-Verlag, 2003.
- [4] Thomas Erl: *Service-Oriented Architecture (SOA). Concepts, Technology, and Design*. Prentice Hall PTR, 2005.
- [5] Dieter Fensel, Holger Lausen, Axel Polleres, Jos de Bruijn, Michael Stollberg, Dumitru Roman, John Domingue: *Enabling Semantic Web Services. The Web Service Modeling Ontology*. Springer, Berlin, Heidelberg, 2006.
- [6] Dieter Fensel: *Ontologies: A Silver Bullet for Knowledge Management and E-Commerce*. Springer, Berlin, Heidelberg, 2. Auflage, 2003.
- [7] Mick Kerrigan, Adrian Mocan, Martin Tanler, Dieter Fensel: *The Web Service Modeling Toolkit - An Integrated Development Environment for Semantic Web Services (System Description)*, 4th European Semantic Web Conference, June 2007, Innsbruck, Austria.
- [8] Adrian Mocan, Emilia Cimpian: An Ontology-Based Data Mediation Framework for Semantic Environments. *International Journal on Semantic Web and Information Systems*, 3(2), 2007, pp. 69 – 98.