1

Semantische Beschreibung von Web Services

Axel Polleres[1], Holger Lausen[1], and Rubén Lara[2]

[1] Digital Enterprise Research Institute (DERI) Innsbruck, Austria and Galway, Ireland, {firstname.lastname}@deri.org[†]
[2] Tecnologia, Informacion y Finanzas (TIF), Madrid, Spanien, rlara@afi.es

Zusammenfassung In diesem Kapitel werden Anwendungsgebiete und Ansätze für die semantische Beschreibung von Web Services behandelt. Bestehende Web Service Technologien leisten einen entscheidenden Beitrag zur Entwicklung verteilter Anwendungen dadurch, dass weithin akzeptierte Standards vorliegen, die die Kommunikation zwischen Anwendungen bestimmen und womit deren Kombination zu komplexeren Einheiten ermöglicht wird. Automatisierter Mechanismen zum Auffinden geeigneter Web Services und deren Komposition dagegen werden von bestehenden Technologien in vergleichsweise geringem Maß unterstützt. Ähnlich wie bei der Annotation statischer Daten im "Semantic Web" setzen Forschung und Industrie grosse Hoffnungen in die semantischen Beschreibung von Web Services zur weitgehenden Automatisierung dieser Aufgaben.

1.1 Einführung und Terminologie

Web Services und Service-orientierte Architekturen sind neue Technologien mit dem Ziel Softwarekomponenten und Geschäftsanwendungen innerhalb von Unternehmen sowie auch über Unternehmensgrenzen hinweg über einheitliche Schnittstellen zugänglich zu machen und damit die einfache Entwicklung verteilter Anwendungen und die Integration bestehender Softwarelösungen zu ermöglichen. Damit haben wir bereits die beiden wichtigsten Anwendungsbereiche von Web Service Technologien umrissen:

- *Integration bestehender Anwendungen (Enterprise Application Integration)*: Die Integration bestehenden Anwendungen, welche oft schon seit Jahren verlässlich im Einsatz sind, gewinnt zunehmend an Bedeutung. Die Entwicklung von Lösungen zur Anwendungsintegatration anstatt kostspieliger Neuentwicklung macht

[†] Die Arbeit der Autoren wird durch die Europäische Kommission im Rahmen der Projekte DIP, Knowledge Web, InfraWebs, SEKT und ASG; durch die Science Foundation Ireland im Rahmen des DERI-Lion Projektes; sowie durch FIT-IT (Forschung, Innovation, Technologie - Informationstechnologie) im Rahmen der Projekte RW^2 und TSC gefördert.

heute einen großen und stetig wachsenden Anteil im Bereich der Informationstechnologie aus.

- *Elektronischer Handel (eCommerce)*: Mit existierenden Web Schnittstellen, die diverse Anbieter ihren Kunden zur Verfügung stellen befinden wir uns erst am Anfang der Möglichkeiten eines wachsenden Wirschaftsfaktors. Online-Händler und -Marktplätze können durch die Nutzung von Web Service Technologien entscheidend von den neuen und engeren Integrationsmöglichkeiten ihrer angebotenen Dienste im Netz über einheitliche Softwareschnittstellen profitieren.

Während Web Services nun einerseits eine technische Grundlage für diese Anwendungen bieten kann andererseits die semantische Beschreibung entscheidend zur weiteren Automatisierung in der Verwendung beitragen.

Wir geben im folgenden nach einer grundsätzlichen Klärung der verwendeten Terminologie einen Überblick über die wichtigsten bestehenden Standards im Bereich Web Services und deren Vorläufer. Anschließend widmen wir uns im Rest des Kapitels den Vorteilen der semantischen Beschreibung von Web Services und präsentieren konkrete vorgeschlagene Ansätze.

Der englische Begriff "Service" ist in seiner Bedeutung überladen [36]. In der Geschäftswelt beispielsweise steht "Service" für eine *Dienstleistung*, das heißt eine Geschäftsaktivität die einen bestimmten Wert für den Kunden hat [8]. In der Informatik wird der Begriff "Service" hingegen oft synonym zum Begriff "Web Service" gesehen, das heißt ein über das Internet abrufbarer elektronischer Dienst in Form einer Softwarekomponente.

Im diesem Kapitel gebrauchen wir diese Begriffe wie folgt:

Service. Ein Service wird von Preist [36] als die Bereitstellung eines Wertes in Form eines Produkts oder einer Dienstleistung verstanden. Nehmen wir zum Beispiel einen Flug von Madrid nach Innsbruck. Im Sinne unserer Definition ist mit Service hier der tatsächliche Transport gemeint. Die Bereitstellung des Service bzw. der Abschluss eines Vertrages ist hier nur bedingt vom Kommunikationsweg abhängig: Um ein Ticket zu erhalten kann gleichermaßen Telefon, Reisebüro oder Website benutzt werden.

Web Service. Web Services werden von Preist als Softwarekomponenten definiert, welche unter Verwendung von Web Service Standards abrufbar sind. Unserem Beispiel folgend, können wir hierin etwa eine Softwarekomponente verstehen, welche die Flugbuchung ermöglicht. Ein Web Service ist also ein Hilfsmittel um einen "Service" in Anspruch zu nehmen, bzw. einen Vertrag über dessen Inhalt abzuschließen.

Service-Orientierte Architektur (SOA). Ein weiterer oft im Zusammenhang mit Web Services genannter Begriff ist die sogenannte Service-Orientierte Architektur. Der Begriff "Service" wird im Zusammenhang mit Service-Orientierten Architekturen oft synonym mit "Web Service" verwendet. Allerdings ist ein Service in einer SOA nicht notwendigerweise ein Web Service im engeren Sinn, sondern bezeichnet generell Softwarekomponenten, die über ein beliebiges Protokoll verfügbar sind. Plattform- und Programmiersprachenunabhängigkeit sind keine notwendigen Kriterien.

1.2 Vorläufer der Web Service Technologie

Das Bestreben nach der Entwicklung von Standards zur Erleichterung der Erstellung verteilter Anwendungen ist in der Informatik nicht neu. Technologien, die eine Abstraktionsschicht zum eigentlichen Kommunikationsmedium bieten, werden im allgemeinen als "Middleware" bezeichnet.

Zahlreiche Bemühungen zur Einführung standardisierter Middlewarelösungen in der Vergangenheit sind indes entweder gänzlich gescheitert oder blieben zumindest in ihrer Anwendung auf unternehmensinterne Lösungen beschränkt, da keine herstellerübergreifende Einigung zur Standardisierung erfolgte.

Die erste Technologie zum Aufruf von Prozeduren in verteilten Systemen mit weiterer Verbreitung stellen Remote Procedure Calls (RPC) dar, zu Beginn der 80er Jahre von Birell und Nelson [13] eingeführt. Zusammen mit einer einheitlichen Interface-Beschreibungssprache (IDL) bietet RPC einen plattform- und programmiersprachenunabhängigen Mechanismus zur Einbettung verteilter Prozeduraufrufe in Applikation. Die in realen Anwendungen gestellten Anforderungen und weite Akzeptanz des objektorienten Paradigmas führten zu zahlreiche Erweiterungen von RPC, wie Transactional RPC, Object Brokers und Remote Method Invocation (RMI), Transaction Processing (TP) Monitors, sowie Message-orientierter Middleware.

Umfassende Middleware-Architekturen wie CORBA – zu Beginn der 90er Jahre von der Object Management Group (OMG) eingeführt – erlaubten bereits dynamische Selektion und Bindung von Services, erfuhren jedoch nicht den gewünschten Verbreitungsgrad und Akzeptanz, um das eigentliche Ziel einer einheitlichen Middleware-Plattform zu erreichen: Effektive, einfache Kommunikation verteilter Softwarekomponenten zur Lösung des Integrationsproblems innerhalb einer Organisation einerseits und über Organisationsgrenzen hinweg.

Die Technologien, die heute unter dem Begriff Web Services zusammengefasst werden haben einige Vorteile gegenüber ihren Vorgängern, welche den Erfolg dieser Technologien zumindest wahrscheinlicher machen:

- Die Kommunikation zwischen Web Services erfolgt über weit verbreitete und unterstützte Protokolle wie HTTP.
- SOAP bietet ein Protokoll zum Austausch von Nachrichten zwischen Web Services, wobei XML als einheitliches Datenformat verwendet wird, was erheblich zur Erleichterung der syntaktischen Interoperabilität von Web Services beiträgt.
- Ähnlich wie schon für RPC und CORBA existiert mit WSDL eine Interface-Beschreibungssprache zur Beschreibung von Operationen und Nachrichten, die vermehrt Akzeptanz findet.
- Im Gegensatz zu einer umfassenden, komplexen Infrastruktur, welche alle Probleme des Zusammenspiels verteilter Applikationen auf einmal in Angriff nimmt, besteht der "Web Services Technology Stack" aus einer modularen Familie von Standards, die jeweils Teilprobleme in Angriff nehmen.
- Im Unterschied zu Vorgängertechnologien scheinen sich die "Global Player" großteils einig geworden zu sein, dass die Lösung von Interoperabilitätsproblemen nicht in Dutzenden proprietären Ansätzen zu finden ist, sondern man arbeitet an gemeinsamen Standard-Empfehlungen.

Einen ausgezeichneten Überblick über Vorläufertechnologien sowie den jetzigen Stand "syntaktischer" [3] Web Service Standards bieten beispielsweise Alonso et al. [4].

1.3 Web Service Technologien im Detail

Web Service Standards im engeren Sinne bauen auf vier Hauptkomponenten auf:

- Ein vereinbartes *Transportprotokoll*.
- Eine plattformunabhängiges *Format zur Beschreibung von Nachrichten* und deren Inhalt.
- Eine *Sprache zur Interface-Beschreibung*, die beschreibt welche Operationen mit welchen Nachrichten ein Service zur Verfügung stellt.
- Ein *Verzeichnis* um verfügbare Services zu publizieren und zu finden.

Die erste Komponente kann im Prinzip durch jedes der verbreiteten Transportprotokolle wie beispielsweise SMTP oder FTP realisiert werden. Das populärste Protokoll im Kontext von Web Services ist HTTP. HTTP hat den Vorteil, dass es zum einen die Nutzung bestehender Infrastruktur erlaubt und zum anderen auch auf vorhandenes Know-How seitens der potenziellen Nutzer aufgebaut werden kann.

Die zweite Komponente wird durch SOAP [42] realisiert. SOAP ist eine Spezifikation zur Vereinheitlichung der Übertragung von XML-codierten Nachrichten. Außerdem spezifiziert SOAP die Bindung an HTTP als unterliegendes Kommunikationsprotokoll zwischen zwei adressierbaren Service-Partnern ("Endpoints"). Kommunikation vie SOAP über HTTP stellt in gewisser Hinsicht eine Lösung bestehender Probleme mit Ansätzen wie RMI und CORBA dar, welche eine enge Koppelung zwischen einzelnen Komponenten festlegen und ausserdem für eine offene "Web Infrastruktur" nicht geeignet sind: Firewalls blockieren in aller Regel die zu Grunde liegenden proprietären Protokolle.

Als dritte Komponente dient die Web Service Description Language (WSDL) [14] zur XML-basierten Beschreibung der Benutzerschnittstelle, welche ein Web Service zur Verfügung stellt. WSDL erlaubt eine getrennte Beschreibung der abstrakten Funktionalität eines Service und den Details wie auf das Service zugegriffen werden kann. Einerseits werden die Operationen eines Service mit ihren Eingabe- und Ausgabeparametern beschrieben, andererseits kann unabhängig davon beschrieben werden über welche Adressen ("Endpoints") das Service aufrufbar ist und welche Transportprotokolle benutzt werden.

Universal Description, Discovery and Integration (UDDI) [12] stellt als Verzeichnistechnologie die Schnittstelle zum Auffinden existierender Web Services die vierte Hauptkomponente Web-Service-basierter Middleware-Lösungen dar. UDDI bietet eine Schnittstelle, über die Geschäftspartner und extern angebotene Services dynamisch gefunden werden können. Ein UDDI Server ist selbst über eine Web Service Schnittstelle via SOAP zugänglich und bietet Operationen zum Publizieren und Suchen registrierter Services. Die Beschreibung eines Service wiederum kann einen Link zu einer bestehenden WSDL Beschreibung beinhalten. Dennoch sind die

[3] Wir sprechen hier von "syntaktischen" Standards, da diese Technologien noch nicht die semantische Annotation von Services behandeln.

Möglichkeiten von UDDI als Verzeichnis zur dynamischen Einbindung bestehender Web Services in verteilte Anwendungen beschränkt. Mit WSDL Beschreibungen stehen Information in maschinenlesbarer Form zur Verfügung jedoch beschreiben diese den Service nur syntaktisch und nicht semantisch. Somit ist zwar das Interface aber nicht die tatsächlich angebotene Funktionalität maschinenlesbar. UDDI ist auf Stichwortsuche in natürlichsprachlichen Beschreibungen der Web Services beschränkt. Die Idee, Beschreibungen mit Semantic Web Technologien zu erweitern welche es erlauben unter Verwendung von Ontologien formal auszudrücken, was ein Web Service leisten kann, liegt nahe.

Eine Reihe weiterer Standards und Empfehlungen komplettieren die derzeit unter dem Begriff "Web Services" zusammengefassten Technologien. Die bisher beschriebenen Komponenten erlauben lediglich die Beschreibung von Services die simplen Interaktionsmustern folgen: WSDL-Beschreibungen sind auf Mengen einfacher Eingabe-/Ausgabe Operationen beschränkt. Allerdings sind Geschäfts-Prozesse oft komplex und erfordern die Beschreibung komplizierterer Interaktionsmuster.

Die Business Process Execution Language for Web Services (BPEL4WS) [6] erlaubt aufbauend auf WSDL die Beschreibung solcher Geschäftsprozesse und damit auch die Beschreibung ausführbarer Kombinationen verschiedener Web Service Aufrufe. BPEL stellt eine XML-basierte Sprache zur Beschreibung und Ausführung von Prozessen dar, welche WSDL Operationen als Einzelaktivitäten betrachten. Der beschriebene Gesamtprozess kann wiederum als eigenständiger Web Service verfügbar gemacht werden.

Weitere Standards zur Lösung der Problemstellungen rund um Web Services, oft unter der gemeinsamen Bezeichnung "WS-*" Standards firmierend, werden derzeit vom W3C[4] und anderen Konsortien e.g. OASIS[5] entwickelt. Ohne auf alle diese erschöpfend eingehend zu wollen, seien einige erwähnt:

- WS-Security [31] ist ein Standard zur Erweiterung von SOAP Nachrichten um Security-Informationen.
- WS-Policy [9] ist ein Beschreibungsformalismus für von einem Web Service unterstützte oder geforderte Richtlinien bezüglich Sicherheit, Sprache, etc.
- WS-PolicyAttachment [10] erlaubt WS-Policy Zusicherungen in WSDL zu verlinken.
- WS-Trust [5] erlaubt die Einrichtung und Verifizierung von Trust Beziehungen zwischen Geschäftspartnern.

1.4 Semantische Servicebeschreibung

Aktuelle Beschreibungsstandards für Web Services haben einen entscheidenden Nachteil: Sie beschränken sich auf die überwiegend syntaktische Beschreibung *wie* Web Services aufzurufen sind, anstatt die maschinenverständliche Formalisierung davon *was* das Service leistet zu ermöglichen. Wir sehen uns bei der Suche nach geeigneten Web Services ähnlichen Hürden gegenüber wie beim Filtern der unüberschaubaren Menge an Daten im "statischen" Web. Beschreibungen mittels Technologien des Semantic Web sind auch hier die Hoffnungsträger.

[4] http://www.w3.org/
[5] http://www.oasis.org/

Die Anwendungsszenarien für semantische Beschreibungen von Web Service Technologien entsprechen grundsätzlich denen normaler Web Service Technologien, mit dem Unterschied, dass die Verwendung von Ontologien und semantischer Beschreibungen von Services genauso wie bei der Annotation statischer Daten einen höheren Grad der Automatisierung verspricht.

Die semantische Beschreibung von Services in einer Art und Weise welche tatsächlich die *automatisierte* Benutzung und Wiederverwendbarkeit von Web Services ermöglicht ist kein einfaches Unterfangen und wir stehen auf diesem Gebiet erst am Anfang. Zum jetzigen Zeitpunkt ist noch nicht klar welche Strategie letztendlich der Technologie zum Durchbruch verhelfen wird und wir beschränken uns hier auf die Vorstellung und den Vergleich der existierenden Ansätze.

Die Bereitstellung einer einheitlichen Kommunikationsinfrastruktur zwischen Web Services eröffnet neue Möglichkeiten: Wie der menschliche Entwickler sollen Maschinen selbst geeignete Web Services dynamisch finden und aufrufen können um eine gestellte Aufgabe zu lösen. Die Entwicklung konzentriert sich hier auf die Standardisierung der semantischen Beschreibung, d.h. **was** ein Service leistet, um diese dynamischen Arbeitsschritte automatisierbar zu machen. Im Rahmen dieses Buches können wir keine erschöpfende Aufzählung aller hierfür zu lösenden Problemstellungen und damit verbundenen Optimierungsmöglichkeiten geben, jedoch stellen wir jene vor, bei denen von semantischer Technologie das größte Potenzial erwartet wird:

Discovery. Bevor ein Web Service in einer verteilten Anwendung genutzt werden kann, muss dieser dem Entwickler, bzw. im automatisierten Fall dem Softwaresystem bekannt sein. Derzeitige Technologien (z.B. UDDI) unterstützen diesen Arbeitsschritt lediglich mittels Stichwortsuche und standardisiertem Vokabular wie UNSPC[6]. Semantische Annotation ermöglicht die Beschreibung von Services mittels dezentraler Ontologien, die über logische Axiome verbunden sind, mittels welcher Inferenzmaschinen berechnen können, welche Services zu einer bestimmten Anfrage passen.

Negotiation. Ist ein geeignetes Web Service gefunden, das zur Lösung des gestellten Problems verwendet werden kann, geht es daran aus der Menge der möglichen Dienste, die dieser Web Service bereitstellt, eine konkrete Serviceinstanz zu bilden. Dies heißt z.B. verschiedene Transport- oder Zahlungsmodalitäten auszuhandeln.

Composition. Im Falle, dass ein Anfrage nicht direkt von einem verfügbaren Web Service bearbeitet werden kann, ermöglichen semantische Beschreibungen die Kombination von mehreren Web Services.

Invocation. Ist ein Service oder eine Kombination von Services gefunden und ausgewählt, kann diese/s ausgeführt werden. Dazu müssen die Information aus den Wissensdatenbanken (z.B. Eingabewerte die in in der semantischen Anfragebeschreibung enthalten sind) auf die von dem jeweiligen Kommunikationsprotokoll geforderten Formate angepasst werden.

[6] United Nations Standard Products and Services Code `http://www.unspsc.org/`

1.5 Ein konkretes Anwendungszenario

Um die Anwendung von Semantic Web Technologien im Bereich Web Services zu illustrieren wählen wir einen Anwendungsfall aus dem europäischen Forschungsprojekt SWWS[7], eines der ersten Forschungsprojekte in Europa mit dediziertem Fokus auf Semantische Web Services. Der vorgestellte Anwendungsfall [18] wurde mit den im Projekt erarbeiteten Technologien gelöst. In diesem Szenario geht es um die

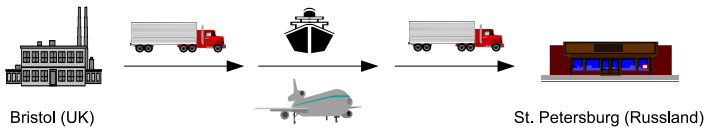

Abbildung 1.1. Anwedungsfall Semantischer Web Services im Logistik Bereich

Durchführung von Warentransporten. Der Transport könnte zwar vollständig von einem externen Dienstleister abgewickelt werden, dies ist jedoch wesentlich teurer als die Summe der Einzeldienstleistungen. Abbildung 1.1 zeigt die einzelnen Abschnitte eines Warentransport. Um eine Ware von Bristol (England) bis St. Petersburg (Russland) zu versenden, muss diese zuerst auf dem Landweg zum nächsten Umschlagplatz gebracht werden, von dort dann alternativ mit dem Flugzeug oder Schiff aufs europäische Festland und schliesslich wieder auf dem Landweg bis zum entgültigen Bestimmungsort. Jeder Teilabschnitt wird über elektronische Datenübertragung (das heisst mit Hilfe von Web Service Aufrufen) initiert und abgewickelt. Innerhalb von SWWS wurden die entsprechenden Web Services semantisch annotiert, um den Prozessablauf während der Abwicklung einer komplexen Transportkette zu optimieren. Der Versender kontolliert hierbei alle Teile der dynamischen Logistikkette selbst (Composition), dass heisst für jede Teilstrecke sucht er passende Serviceanbieter aus (Discovery), verhandelt mit den geeigneten Anbietern (Negotiation), wählt schliesslich den geeignetsten aus und führt mit diesem den Transport aus (Invocation). Bei der semantischen Beschreibung geht es nun darum, geeignete Ontologien zu entwickeln, die es den einzelnen Serviceanbietern ermöglichen ihr Service entsprechend zu bewerben, das heisst z.B. eine formale Beschreibungsmethode zu definieren, die es erlaubt in maschinenlesbarer und -analysierbarer Form Sachverhalte auszudrücken wie "Web Service S_1 erlaubt die *Buchung* einer *Schiffslieferung* von *London* nach *Helsinki*, mit gegebener *Gewichts- und Größenbeschränkung* zu einem gegebenen, *gewichtsabhängigen Tarif*". Diese Beschreibung soll es ermöglichen, dass ein geeigneter Algorithmus Sevice s_1 als geeignet identifiziert, einen Teilabschnitt des Transportproblems zu übernehmen, sowie die damit verbundendn Kosten zu kalkulieren und mit anderen Varianten zu vergleichen, etc.

1.6 Ansätze und Beschreibungsmethoden

Im folgenden geben wir einen Überblick über die drei wichtigsten derzeit vorgeschlagenen Ansätze zur semantischen Beschreibung von Web Services. Wir beschränken

[7] Semmantic Web Enabled Web Services `http://swws.semanticweb.org/`

uns hier auf die konzeptuelle Beschreibung und verzichten auf Code-Beispiele, interessierte Leser seien hier auf die entsprechenden Spezifikationen verwiesen. Für alle drei Ansätze wird ein kurzer Überblick über Tool-Support und die Unterstützung automatisierter Discovery, Negotiation, Composition und Invocation gegeben.

1.6.1 OWL-S

OWL-S [28] war die erste Initiative zur Definition einer Standard-Ontologie zur semantischen Annotation von Web Services. Seit der ersten Veröffentlichung des Vorgängers im Mai 2001 unter dem Namen DAML-S bis zur aktuellen Version 1.1, welche im September 2004 von Nokia, University of Maryland, dem National Institute of Standards and Technology (NIST), Network Inference, SRI International, France Telecom, Stanford University, Toshiba, und der University of Southampton als Standardvorschlag beim W3C eingereicht wurde hat OWL-S zahlreiche Verbesserungen und Weiterentwicklungen erfahren. Im Grunde wurden aber die Eckpfeiler der Ontologie von Anfang an aufrecht erhalten.

OWL-S baut einerseits auf bestehenden Web Service Standards auf, indem es die Anbindung an bestehende WSDL Beschreibungen erlaubt, andererseits auf Semantic Web Standards, indem bestehende OWL oder RDFS Ontologien zur Beschreibung von Services und deren Paramtetern genutzt werden können und OWL-S wiederum wie der Name bereits andeutet selbst als Ontologie in der Web Ontology Language (OWL) beschrieben ist. Diese Ontologie benutzt drei Hauptkonzepte zur Beschreibung eines Web Service:

Service Profile. Das Profil beschreibt abstrakt was ein Service tut, seine Funktionalität und andere nicht-funktionale Aspekte die zur Aufffindung des Service nützlich sein können. Ein Profil hat Attribute zur Beschreibung von Eingabe- und Ausgabeparametern als OWL Konzepte, sowie Vorbedingungen und Effekten welche als logische Ausdrücke in einem bestimmten Wissensrepräsentationsformalismus beschrieben werden können. Desweiteren erlaubt OWL-S die Zuordnung zu einer bestimmten Kategorie (serviceCategory), wodurch eine externe Taxonomie von Service Profilen referenziert werden kann. Daneben existieren nichtfunktionale Attribute zur Beschreibung des Serviceanbieters und zur Servicequalität.

Service Modell. Das Service Modell beschreibt wie ein Service seine Funktionalität erreicht, konkret erlaubt das OWL-S Service Modell die Beschreibung eines Service als Prozess. Dazu unterscheidet OWL zwischen atomicProcess (atomares Service), compositeProcess (Prozessenkomposition) und simpleProcess (simpler Prozess). AtomicProcesses erlauben wie schon das Service Profile die Beschreibung von Eingabe- und Ausgabeparametern, Vorbedingungen, Effekten, sowie bedingten Ausgaben. CompositeProcesses erlauben es, einzelne Aktivitäten, die wiederum durch OWL-S simpleProcesses oder atomicProcesses beschrieben werden können, mit Konstrukten sehr ähnlich denen in BPEL4WS in einem Workflow zu kombinieren. Ein simpleProcess stellt eine abstrakte Sicht auf einen compositeProcess dar. Damit bietet OWL-S ein vollständiges Modell zur Beschreibung von Servicekompositionen.

Service Grounding. Das Grounding beschreibt, wie ein Service benutzt und aufgerufen werden kann. Obwohl sich OWL-S nicht auf die Beschreibung von Web Services mit einer vorhandendn WSDL Beschreibung festlegt, ist die Anbindung an eine WSDL-Beschreibung derzeit die einzige definierte Methode.

Die formale Beschreibung von Bedingungen und Effekten in OWL-S Profilen und Modellen ist in verschiedenen Wissensrepräsentationssprachen möglich. Konkret werden SWRL [20], KIF [1] oder DRS [29] vorgeschlagen. Die Kombination dieser Formalismen mit den OWL zu Grunde liegenden Beschreibungslogiken [7] hingegen ist noch nicht vollständig geklärt. Ansätze zur Nutzung von OWL-S Beschreibungen in konkreten Implementierungen beschränken sich meist auf Subsumptions-Abfragen von Ein- und Ausgabeparametern im Service Profile basierend auf Inferenzmaschinen für Beschreibungslogiken. Uns sind keine Tools oder Implementierungen bekannt die mit vollen SWRL, KIF oder DRS Beschreibungen arbeiten.

OWL-S ist mehr eine Meta-Ontologie für Web Services in OWL denn eine eigenständige Beschreibungssprache. OWL-S ist somit zwar eine OWL Ontologie, dennoch beinhalten vollständige OWL-S Beschreibungen weit mehr an semantischer Information, als durch die eigentliche Semantik der OWL zu Grunde liegenden Beschreibungslogik ausgedrückt wird. Dies wurde auch vom OWL-S Konsortium selbst erkannt und es wurde einerseits versucht die Semantik von OWL-S Modellen formal zu beschreiben [32] andererseits arbeitet man derzeit an einer eigenen abstrakten Syntax für OWL-S Beschreibungen von Services, die nicht an das enge Korsett von OWL gebunden ist. Das Semantic Web Services Framework (SWSF) [11], eine Initiative, welche in gewissem Sinne als Nachfolger von OWL-S gesehen werden kann, versucht alternativ dazu eine völlig eigenständige Definition der Semantik der OWL-S Konzepte in einer Ontologie, die nicht in OWL sondern direkt in Prädikatenlogik erster Stufe formalisiert ist. Da SWSF allerdings bislang eher von rein theoretischer Bedeutung ist und noch keinerlei Implementierungen basierend auf SWSF existieren werden wir diesn Ansatz hier nicht gesondert behandeln.

Discovery. Das Element in OWL-S zur Bewerbung von Web Services in Verzeichnissen und deren Auffindung ist das Service Profil. Vorschläge zur Einbettung von OWL-S Profilbeschreibungen in UDDI existieren in der Literatur [40]. Um die semantischen Beschreibungen tatsächlich nutzen zu können muss die derzeit relativ einfache Abfrageschnittstelle von UDDI mit einer Inferenzmaschine erweitert werden.

Negotiation. Um ausreichende Information für das abschliessen eines "Servicevertrags" zu beschreiben bietet OWL-S die nichtfunktionalen Attribute im Service Profil. Kombinationen mit Standards wie WS-Policy etc. werden in der derzeitigen Version nicht beschrieben.

Composition. Es existieren Vorschläge um Semantische Discovery in OWL mit Planungsansätzen aus der Künstlichen Intelligenz [39, 27] zu kombinieren. Diese Planer nutzen aufgrund der Problemkomplexität allerdings oft nicht die gesamte Ausdrucksstärke des OWL-S Prozessmodells.

Invocation. Das Element in OWL-S zur Beschreibung des Aufrufs eines Service ist das Service Grounding. Konkret erlaubt das Grounding die Anbindung von atomaren Prozessen aus dem Service Modell an WSDL Operationen, wobei einzelne Eingabe- und Ausgabeparamter an WSDL Eingabe- und Ausgabenachrichten gebunden werden.

Tool Support. Ohne Anspruch auf Vollstänfigkeit seien hier einige Implementierungen basierend auf OWL-S genannt: Der OWL-S Matchmaker [34] ist ein System für Service Discovery, welches auch in UDDI integriert wurde. Mindswaps OWL-S

API [38] stellt eine Java Schnittstelle für OWL-S Beschreibungen zur Verfügung. Das API erlaubt das Parsen, Serialisieren, und die Ausfürung von OWL-S Services. Der OWL-S Editor [17] ist ein Plug-in für den Ontologie Editor Protégé und erlaubt das graphische Editieren von OWL-S Beschreibungen.

1.6.2 WSMO

Die Web Service Modeling Ontology (WSMO) [16] – im April 2005 als Standardvorschlag beim W3C eingereicht – ist einerseits konzeptuelles Modell für die semantische Servicebeschreibung, und bietet andererseits eine vollständige Beschreibungssprache, WSML, zur Annotation von Web Services. Die WSMO Arbeitsgruppe wurde Anfang 2004 gegründet und ist eine vorwiegend europäische Initiative der EU-Projekte SEKT [8], DIP [9] und KnowledgeWeb [10].

WSMO verfolgt im Prinzip ähnliche Grundsätze wie OWL-S allerdings mit einem etwas anderen Fokus. Wie bei OWL-S sind Ontologien ein wichtiges Element. Allerdings ist WSMO nicht selbst als OWL Ontologie, sondern in einem Meta-Datenmodell – der MOF (Meta-Object Facility) Methodologie [2] folgend – formalisiert. Dieses Metamodell kann in verschiedenen Wissensrepräsentationssprachen ausgedrückt werden.

WSMO folgt dem Grundprinzip der strikten Entkoppelung von Entitäten, das heißt Servicebeschreibungen, Ontologien und Benutzeranfragen sind unabhängige Entitäten im Web welche mittels Mediatoren miteinander in Beziehung gesetzt werden. Demzufolge unterscheidet WSMO vier zur Beschreibung von Web Services relevante Hauptelemente:

Ontologien definieren das Vokabular um alle anderen Elemente wie Services und Anfragen zu beschreiben. Ontologien werden in WSML (Web Service Modeling Language), einer Sprache die speziell für WSMO entwickelt wurde, formalisiert. WSML subsumiert die Ausdrucksstärke von OWL, folgt allerdings eher einem Frame-basierten Ansatz. Zugrundeliegende logische Formalismen zur Beschreibung von Axiomen und Regeln in Ontologien basieren im Gegensatz zu OWL nicht auf Beschreibungslogiken, sondern auf Frame-Logik [24] und Logischer Programmierung [26].

Web Services werden in WSMO aus drei verschiedenen Perspektiven beschrieben: Nichtfunktionale Aspekte (Nonfunctional Properties), funktionale Aspekte (Capability), und dynamische Aspekte (Interfaces), das heißt das Verhalten des Web Service. Capabilities erlauben es, ähnlich wie in OWL-S, Vorbedingungen, Annahmen, Nachbedingungen und Effekte mittels logischer Ausdrücke spezifizieren. Bei der Interfacebeschreibung unterscheidet WSMO zwischen Choreopgraphy Interface, das heißt der Schnittstelle zum Benutzer und Orchestration Interface, das heißt, welche Services und Goals ein Web Service aufruft. Im weitesten Sinne kann man Choreography und Orchestration Interfaces mit dem OWL-S Prozessmodell vergleichen. Ähnlich dazu erlaubt WSMO die Anbindung an bestehende WSDL Beschreibungen über einen Grounding-Mechanismus.

[8] `http://www.sekt-project.com/`
[9] `http://dip.semanticweb.org`
[10] `http://knowledgeweb.semanticweb.org/`

Goals spezifizieren Anfragen, dass heißt die benötigte Funktionalität aus der Benutzerperspektive. Goals stellen ein konzeptuelles Spiegelbild von Servicebeschreibungen dar. Die Entkopplung von Goal- und Servicebeschreibungen als eigene Entitäten (*goal-driven approach*) ist ein entscheidender Unterschied zu OWL-S.

Mediatoren beschreiben Elemente zur Überwindung von Heterogenitäten zwischen verschiedenen Komponenten. Mediatoren lösen Inkompatibilitäten auf verschiedenen Ebenen:

- Datenebene – Mediatoren definieren Regeln zur Auflösung von statischen Terminologiekonflikten [37].
- Prozessebene – Mediatoren lösen Konflikte aufgrund von unterschiedlichen Interaktionsmustern durch Web Services beschriebener Prozesse [15].

Mediatoren in WSMO können die verschiedenen anderen Elemente (**O**ntologien, **W**eb Services und **G**oals) verbinden, demzufolge unterscheidet man zwischen **OOMediator**, **WWMediator**, **WGMediator** und **GGMediator**, wobei ersterer ein reiner Datenmediator ist und die anderen drei auch Prozessmediation beinhalten können.

WSML baut ähnlich wie OWL auf bestehenden Standards auf und bietet sowohl XML als auch RDF Serialisierung. Die formale Semantik von WSML ist noch nicht vollständig für alle Elemente von WSMO definiert.

Discovery. Discovery in WSMO wird durch Vergleich von Goal- und Servicebeschreibungen erreicht. Ein mehrschichtiges Modell für semantische Discovery in WSMO [21] wurde zum Beispiel in einer Implementierung basierend auf Logischer Programmierung implementiert [23]. Weitere Implementierungen für Discovery Engines in WSMO existieren im COCOON-Projekt [11] und in WSMX[22].

Negotiation. Wie in OWL-S können Policies in WSMO zu einem gewissen Grad mittels nichtfunktionaler Attribute des Web Service beschrieben werden. Abgesehen davon existieren Arbeiten zur Integration von Peer Trust Policy Regeln in WSMO [33].

Composition. Das Element in WSMO zur Beschreibung von komplexen Prozessen sind Orchestration Interfaces. Derzeit sind allerdings noch keinerlei Implementierungen in WSMO vorhanden. WSMO legt sich bisher nicht auf eine konkrete Beschreibungssprache für Servicekompositionen fest, lediglich ein konzeptionelles Modell basierend auf Abstract State Machines ist definiert.

Invocation. Ähnlich wie in OWL-S wird in WSMO derzeit ein Grounding-Mechanismus für WSDL Beschreibungen entwickelt. Die Anbindung an WSDL erfolgt innerhalb der Choreography und Orchestration des Web Service. Es ist allerdings nicht ausgeschlossen, dass dieser Grounding-Mechanismus letztendlich von einem an WSDL-S (siehe nächster Abschnitt) angelehntem Modell ersetzt wird.

Tool Support. Derzeit existieren zwei Ausführungsumgebungen für WSMO-kompatible Web Services: (1) WSMX[12], von der WSMO/L/X Arbeitsgruppe als Open Source Initiative entwickelt und (2) IRS-III [30] von der Open University in England. Ein Java API, WSMO4J[13], stellt Werkzeuge zum Parsen, Validieren und zur Serialisierung von WSML-Beschreibungen, sowie Unterstützung von Inferenzen durch Anbindung an verschiedene Inferenzmaschinen zur Verfügung.

[11] `http://cocoon.cefriel.it/RD2/usecases/semantic-discovery-of-cop`
[12] `http://wsmx.sourceforge.org`
[13] `http://wsmo4j.sourceforge.net`

1.6.3 WSDL-S

WSDL-S [3] wurde im November 2005 von IBM und der University of Georgia beim W3C eingreicht. Der Vorschlag stellt einen Bottom-up Ansatz basierend auf WSDL dar und ergänzt damit bestehende Web Service Interface-Beschreibungen mit semantischer Information. Dieser "leichtgewichtige" Ansatz bietet begrenztes Potential zur Automatisierung von Prozessen (vorwiegend für einfache Interaktionsmodelle).

Die WSDL-S Philosophie ist es soweit wie möglich auf bestehenden und akzeptierten Web Service Standards aufzusetzen um möglichst schnell Akzeptanz und Resultate vorweisen zu können. Die Eckpunkte lassen sich wie folgt zusammenfassen:

1. WSDL-S ist direkt in bestehende Standards (WSDL) integriert.
2. Bezüglich Annotationsmechanismen legt sich WSDL-S nicht auf eine konkrete Sprache zur Wissensrepräsentation fest: Verschiedene Wissensrepräsentationformalismen sollten für die Annotierung eines Service zugelassen werden.
3. Vorhandene Typisierungen der Serviceparameter mittels XML Schema wie derzeit von WSDL unterstützt sollen genutzt und mit der semantischen Beschreibung integriert werden.
4. Ein Mechanismus zur Umwandlung zwischen syntaktischer Typisierung mittels XML und den Konzepten einer Ontologie sollte bereitgestellt werden.

WSDL-S fügt hierzu eine kleine Anzahl zusätzlicher Elemente zum bestehenden WSDL Standard hinzu. Diese Elemente erlauben es die Eingabe- und Ausgabeparameter, sowie die WSDL-Operationen an sich zu annotieren. Vor- und Nachbedingungen für Operationen beschreiben semantisch Anforderungen an den Zustand der Welt vor und nach Ausführung einer Operation, die erfüllt werden müssen beziehungsweise garantiert werden. Zusätzlich besteht die Möglichkeit WSDL 2.0 Port Types entsprechend einer Ontologie zu kategorisieren.

Das zweite Grundprinzip hat dazu geführt das die semantischen Modelle nicht direkt im WSDL abgelegt sondern lediglich referenziert werden. Hierbei werden Konzepte in bestehenden Ontologien mittels ihrer URI angegeben. WSDL-S legt dabei nicht fest wie Ontologien definiert werden.

Die Integration von XML Schema wurde insofern realisiert als das zwei Alternativen zur Annotation angeboten werden: Entweder können komplexe Typen oder direkt die einzelnen Unterelemente eines komplexen Typen annotiert werden.

Als Mechanismus für die Umwandlung von Informationen aus Ontologien (zum Beispiel in OWL) und deren Äquivalent in XML Schema werden Referenzen zu Übersetzungsregeln (in XSLT) genutzt.

Discovery. Die Beschreibungen von Vor- und Nachbedingungen erlaubt wie bei den anderen beschriebenen Ansätzen automatisierte Discovery. In [41] wird ein P2P-Infrastruktur präsentiert, die WSDL-S als Beschreibungsformalismus nutzt. Ähnlich zur OWL-S basierenden Discovery erfolgt auch diese auf der Grundlage von Subsumptionsbestimmung zwischen Konzepten in einer Beschreibungslogik.

Negotiation. Als leichtgewichtigerÄnsatz bietet WSDL-S keine explizite Unterstützung für Negotiation. Allerdings kann WSDL-S als Erweiterung von "purem" WSDL evtl. einfacher als andere Ansätze mit entsprechenden Web Service Standardempfehlungen wie WS Policy kombiniert werden.

Composition. WSDL-S verfügt nicht über die Ausdruckskraft um Prozesse zu beschreiben und auch die Beschreibung komplexer Interaktionsmuster ist in der Konzeption des Ansatzes nicht vorgesehen. Allerdings können mit den gegebenen Informationen in Vor- und Nachbedingungen implizite Abfolgen von WSDL Operationen ausgedrückt werden.

Invocation. Als Erweiterung von WSDL bietet WSDL-S unterützung ür automatische Web Service Ausführung basierend auf bestehenden Ansätzen zur Ausführung mittels WSDL beschreibener Services. Zur Integration der zusätzlichen semantischen Information ist ähnlich wie bei OWL-S ein Mapping zwischen ontologischen Konzepten und den entsprechenden XML Schema Typen nötig.

Tool Support. Die meisten Tools stammen direkt aus dem METEOR-S Project[14] Unter anderem wurde eine Web Service Discovery Infrastruktur [41] entwickelt und ein Framework zur Annotation von Web Services [35].

1.7 Zusammenfassung und Ausblick

Wir haben in diesem Kapitel nach einer Vorstellung der wichtigsten Web Service Technologien und der Motivation des Nutzens semantischer Beschreibung von Web Services, drei mögliche Ansätze vorgestellt. Obwohl wir die Ansätze und Strategien im Rahmen dieses Buches nicht erschöpfend behandeln konnten, hoffen wir dennoch einen Eindruck des enormen Potentials dieser Methoden zur Automatisierung von Aufgaben wie der Discovery, Negotiation, Composition und Invocation in Service-orientierten Architekturen vermittelt zu haben.

Standardisierungsorganisationen haben die Wichtigkeit semantischer Annotation von Web Services zum vollen Durchbruch von Web Service Technologien erkannt. Das zeigen Initiativen wie die "Semantic Web Services Interest Group"[15], welche Bestandteil der Web Service Aktiviät des World Wide Web Konsortiums (W3C) ist, oder das jüngst von der Organization for the Advancement of Structured Information Standards (OASIS) ins Leben gerufene "Semantic Execution Environment Technical Committee"[16], welches die Weiterentwicklung und Standardisierung der WSMX Technologie als Referenztechnologie für Semantische Web Services innerhalb OASIS zum Ziel hat.

Dennoch sind die Ergebnisse beziehungsweise die Reife zur industrietauglichen Standardisierung entsprechender Technologien im Moment noch nicht voll gegeben, wie etwa eine Tagung des W3C zum Thema "Frameworks for Semantics in Web Services"[17] zeigte: Gründe, die hierfür angeführt werden sind (1) die noch forschungslastigen Ansätze, (2) mangelnde Industrieunterstützung für Toolentwicklung und Beteiligung an konkreten Standardisierungsvorhaben, und (3) keine erkennbare Präferenz der Web Service Community semantische Ansätze aufzugreifen. Nichtsdestotrotz, liegen große Hoffnungen in der Weiterentwicklung und Konvergenz der derzeitigen Ansätze.

[14] `http://lsdis.cs.uga.edu/projects/meteor-s/`
[15] `http://www.w3.org/2002/ws/swsig/`
[16] `http://xml.coverpages.org/OASIS-SemanticEx-CFP.html`
[17] `http://www.w3.org/2005/04/FSWS/workshop-report.html`

Ein weiteres Problem wird von in der grundsäztlichen Inkompatibilität derzeitiger Web Service Technologien mit Semantic Web Technologien gesehen. Web Services versprechen eine global skalierbaren Middlewarelösung: Solange Technologien rund um WSDL und SOAP jedoch wie derzeit weitgehend lediglich als Form von synchronem "RPC über Web Protokolle" misverstanden wird, bewegen wir uns in einer Sackgasse. Alternative Kommunkationsparadigmen basierend auf Message-orientierte Middelware, Blackboards oder LINDA [19] könnten hier eine sinnvolle Ergänzung bieten, da diese mehr den Web-Prinzipien persistenter Publikation und asynchroner Kommunikation entsprechen. Für Details sei hier auf [25] verwiesen.

Literatur

1. KIF. knowledge interchange format: Draft proposed american national standard (dpANS). Technical Report NCITS.T2/98-004, ANSI KIF Ad Hoc Group, 1998. `http://logic.stanford.edu/kif/dpans.html`.
2. Meta-object facility. Technical report, The Object Management Group, 2004. `http://www.omg.org/technology/documents/formal/mof.htm`.
3. R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M. T. Schmidt, A. Sheth, and K. Verma. Web Service Semantics - WSDL-S. W3C Member Submission, `http://www.w3.org/Submission/2005/10/`, Nov. 2005.
4. G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer, 2004.
5. S. Anderson, J. Bohren, T. Boubez, M. Chanliau, G. Della-Libera, B. Dixon, P. Garg, M. Gudgin, P. Hallam-Baker, M. Hondo, C. Kaler, H. Lockhart, R. Martherus, H. Maruyama, A. Nadalin, N. Nagaratnam, A. Nash, R. Philpott, D. Platt, H. Prafullchandra, M. Sahu, J. Shewchuk, D. Simon, D. Srinivas, E. Waingold, D. Waite, D. Walter, and R. Zolfonoon. Web Services Trust Language (WS-Trust). Technical report, Actional, BEA, Computer Associates, IBM, Layer 7, Microsoft, Oblix, Open Network, Ping Identity, Reactivity, RSA Security, Verisign, February 2005.
6. T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. Business Process Execution Language for Web Services version 1.1. Specification, May 2003. `http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/`.
7. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, (eds.) *The Description Logic Handbook*. Cambridge University Press, 2003.
8. Z. Baida, J. Gordijn, B. Omelayenko, and H. Akkermans. A Shared Service Terminology for Online Service Provisioning. In *ICEC04*, Delft, Netherlands, 2004.
9. S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker, M. Hondo, C. Kaler, D. Langworthy, A. Malhotra, A. Nadalin, N. Nagaratnam, M. Nottingham, H. Prafullchandra, C. von Riegen, J. Schlimmer, C. Sharp, and J. Shewchuk. Web Services Policy Framework (WS-Policy). Technical report, BEA, IBM, Microsoft, SAP, Sonic Software, Verisign, Sept. 2004.
10. S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker, M. Hondo, C. Kaler, A. Malhotra, H. Maruyama, A. Nadalin, M. Nottingham, D. Orchard, H. Prafullchandra, C. von Riegen, J. Schlimmer, C. Sharp, and

J. Shewchuk. Web Services Policy Attachment (WS-PolicyAttachment). Technical report, BEA, IBM, Microsoft, SAP, Sonic Software, Verisign, Sept. 2004.
11. S. Battle, A. Bernstein, H. Boley, B. Grosof, M. Gruninger, R. Hull, M. Kifer, D. Martin, D. L. McGuinness, S. McIlraith, G. Newton, D. De Roure, M. Skall, J. Su, S. Tabet, and H. Yoshida. Semantic web services framework (SWSF). W3C Member Submission, `http://www.w3.org/Submission/2005/07/`, May 2005.
12. T. Bellwood, L. Clément, D. Ehnebuske, A. Hately, Maryann Hondo, Y.L. Husband, K. Januszewski, S. Lee, B. McKee, J. Munter, and C. von Riegen. UDDI version 3.0. `http://uddi.org/pubs/uddi-v3.00-published-20020719.htm`, July 2002.
13. A. D. Birell and B. J. Nelson. Implementing remote procedure calls. *ACM Transactions of Computer Systems*, 1(2):39 – 59, 1984.
14. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL) 1.1. `http://www.w3.org/TR/wsdl`, March 2001.
15. E. Cimpian and A. Mocan. Wsmx process mediation based on choreographies. In *1st Int'l Workshop on Web Service Choreography and Orchestration for Business Process Management (BPM 2005)*, 2005.
16. J. de 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, and M. Stollberg. Web service modeling ontology (WSMO). W3C Member Submission, `http://www.w3.org/Submission/2005/06/`, April 2005.
17. D. Elenius, G. Denker, D. Martin, F. Gilham, J. Khouri, S. Sadaati, and R. Senanayake. The OWL-S editor – a development tool for semantic web services. In *Proc. of the 2nd European Semantic Web Conference (ESWC2005)*, Heraklion, Crete, May 2005.
18. J. Esplugas-Cuadrado. B2B demosntrator description. Technical report, Semantic Web Enabled Web Services, Sept. 2004.
19. D. Gerlernter. Generative communication in Linda. *ACM Transactions on Prog. Lang. and Systems (TOPLAS)*, 1(7):80–112, 1985.
20. I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A semantic web rule language combining OWL and RuleML. `http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/`, May 2004.
21. U. Keller, R. Lara, H. Lausen, A. Polleres, and D. Fensel. Automatic location of services. In *Proc. of the 2nd European Semantic Web Conference (ESWC2005)*, Heraklion, Crete, May 2005.
22. M. Kerrigan. Web service selection mechanisms in the web service execution environment (WSMX). In *Semantic-Based Resource Discovery, Retrieval and Composition (RDRC) Track of the 21st Annual ACM Symposium on Applied Computing (SAC'06)*, April 2006. Accepted for publication.
23. M. Kifer, R. Lara, A. Polleres, C. Zhao, U. Keller, H. Lausen, and D. Fensel. A logical framework for web service discovery. In *ISWC 2004 Workshop on Semantic Web Services: Preparing to Meet the World of Business Applications*, Hiroshima, Japan, 2004.
24. M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *JACM*, 42(4):741–843, 1995.
25. R. Krummenacher, M. Hepp, A. Polleres, C. Bussler, and D. Fensel. WWW or What is Wrong with Web services. In *Proc. of the 3rd European Conference on Web Services (ECOWS 2005)*, Växjö, Sweden, Nov. 2005.

26. J.W. Lloyd. *Foundations of Logic Programming (2nd Edition)*. Springer, 1987.
27. Daniel J. Mandell and Sheila A. McIlraith. Adapting bpel4ws for the semantic web: The bottom-up approach to web service interoperation. In *Proc. of the 2nd Int'l Semantic Web Conference (ISWC2003)*, pages 227–241, 2003.
28. David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. OWL-S: Semantic markup for web services. W3C Member Submission, `http://www.w3.org/Submission/2004/07/`, Nov. 2004.
29. D. McDermott. DRS: A set of conventions for representing logical languages in RDF. `http://www.daml.org/services/owl-s/1.1B/DRSguide.pdf`, January 2004.
30. E. Motta, J. Domingue, L. Cabral, and M. Gaspari. Irs-ii: A framework and infrastructure for semantic web services. In *2nd Int'l Semantic Web Conference (ISWC2003)*. Springer Verlag, October 2003.
31. A. Nadalin, C. Kaler, P. Hallam-Baker, and R. Monzillo (eds.). Web Services Security: SOAP Message Security 1.0 (WS-Security 2004). Technical report, OASIS, March 2004.
32. S. Narayanan and S. McIlraith. Simulation, verification and automated composition of web services. In *Proc. of the 11th Int'l World Wide Web Conference (WWW2002)*, Honolulu, Hawaii, May 2002.
33. D. Olmedilla, R. Lara, A. Polleres, and H. Lausen. Trust negotiation for semantic web services. In *1st Int'l Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), in conjunction with the 2004 IEEE Int'l Conference on Web Services (ICWS 2004)*, San Diego, California, USA, 2004.
34. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web services capabilities. In I. Horrocks and J. Handler, editors, *1st Int. Semantic Web Conference (ISWC)*, pages 333–347. Springer Verlag, 2002.
35. A. Patil, S. Oundhakar, A. Sheth, and K. Verma. METEOR-S Web service Annotation Framework. In *Proc. of the 13th Int'l World Wide Web Conference (WWW2004)*, pages 553–562, 2004.
36. C. Preist. A conceptual architecture for semantic web services. In *Proc. of the Int'l Semantic Web Conference 2004 (ISWC 2004)*, Nov. 2004.
37. F. Scharffe and J. de Bruijn. A language to specify mappings between ontologies. In *IEEE SITIS'05*, Yaoundé, Cameroon, Nov. 2005.
38. E. Sirin and B. Parsia. The owl-s java api. Nov. 2004.
39. E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau. HTN planning for web service composition using SHOP2. 1(4):377–396, 2004.
40. N. Srinivasan, M. Paolucci, and K. Sycara. Adding OWL-S to UDDI, implementation and throughput. In *1st Int'l Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, July 2004.
41. K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, and J. Miller. METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services. *Journal of Information Technology and Management, Special Issue on Universal Global Integration*, 6(1):17–39, 2005. Kluwer Academic Publishers.
42. W3C. SOAP version 1.2 part 0: Primer. `http://www.w3.org/TR/2003/REC-soap12-part0-20030624/`, June 2003.