

## Des Pudels Kern

# Semantik, Odem einer Service-orientierten Architektur

Wolfgang Dostal, Mario Jeckle

*Ein wichtiger Schritt in Richtung Service-orientierte Architektur (SOA) sind Web Services. Allerdings fokussieren letztere auf die syntaktischen Aspekte einer SOA und erfüllen deren eigentliche Anforderung „erbringe einen Service“ nicht. Das flexible Lösen einer Aufgabe setzt voraus, dass die Bedeutung (Semantik) der Aufgabe und eines Services eigenständig (automatisch) interpretiert werden kann. Ohne diese Möglichkeit muss jeder Schritt und jeder zu nutzende Service explizit vorgegeben werden. Oder mit anderen Worten „nutze diesen Service jetzt“. Ziel dieser Artikelserie ist es, wichtige Konzepte für das Erfassen von Semantik und Ansätze zur Einbettung in Web Services vorzustellen. Im Rahmen des ersten Artikels wird die Problematik des Fehlens von Semantik in Web Services herausgearbeitet. Für mögliche Lösungsansätze wird das Semantic Web mit seinen semantischen Konzepten vorgestellt.*

► Zum Thema Web Services, als aktuell bevorzugter Technik zur Umsetzung einer Service-orientierten Architektur, werden momentan eine Vielzahl von Vorschlägen, Empfehlungen und Standards diskutiert. Besondere Aufmerksamkeit erfahren dabei die beiden Gebiete Sicherheit und Workflow. Beides sind entscheidende Aspekte, um die versprochene Automatisierung der Kommunikation zwischen Anwendungen über öffentliche Netze zu erreichen. Zur vollständigen Automation muss jedoch die semantische Komponente hinzukommen. Bislang haftet diesem Technikgebiet noch ein Hauch von Esoterik und akademischer Abgehobenheit an, was zumindest aus Sicht der Gartner Group nicht mehr lange der Fall sein wird [GAR].

## Agenda

Unser Ziel mit dieser Artikelserie ist es, zu zeigen, dass die Techniken für „Semantik“ nicht nur was für „Gurus“ sind. Vielmehr ist „Semantik“ eine Notwendigkeit und auch von „normal Sterblichen“ einsetzbar. Wir werden zeigen, dass Semantik *bedeutend* ist\*, von uns allen bereits genutzt wird und dass Web Services als ein Träger von Semantik geeignet sind. Dazu haben wir die Diskussion in vier Artikel aufgeteilt:

▼ **Semantik, Odem einer Service-orientierten Architektur:** Dieser Artikel beschreibt die

\* Was Bedeutung ohne Semantik bedeutet, muss noch gedeutet werden;-)

Motivation, um Semantik in Service-orientierten Architekturen zu nutzen.

▼ **Beschreibung von Semantik:** Im Rahmen dieses Artikels wird vorgestellt, wie mit Hilfe der W3C-Empfehlung Resource Description Framework (RDF) und ergänzender Techniken Semantik standardisiert beschrieben werden kann.

▼ **Taxonomien und Ontologien:** Dieser Artikel beschreibt die Bedeutung dieser Begriffe und deren Nutzen für die Semantik in einer Service-orientierten Architektur.

▼ **Web Services, der natürliche Träger von Semantik:** Nachdem in den vorangegangenen Artikeln die Grundlagen gelegt wurden, wird im abschließenden Artikel gezeigt, wie Semantik in Web Services eingebunden werden kann bzw. könnte.

Der vorliegende Artikel diskutiert die Aussage: Semantik, Odem einer Service-orientierten Architektur. Um diese Aussage zu „beweisen“, werden in dieser Artikelserie konkret Web Services, als derzeit populärstes Konzept einer Service-orientierten Architektur (SOA), betrachtet.

## Service-orientierte Architektur

Die Idee einer Service-orientierten Architektur (SOA) ist kein abstraktes Konstrukt, das mit der Einführung von Web Services entstanden ist. Vielmehr ist das SOA-Konzept bereits in den 90er Jahren als Bestandteil der CORBA-Architektur beschrieben worden. Seitdem wurde dieses Konzept weiter ausgearbeitet und verfeinert.

Kernelement einer Service-orientierten Architektur ist der namensgebende Service (Dienstleistung). Dieser Begriff ist kein ursprünglicher IT-Begriff, und jeder wird schon einmal über die Bedeutung des Begriffs Service nachgedacht haben. Ob bei der Nutzung von öffentlichen Verkehrsmitteln, beim Friseur oder dem Abgeben der Steuererklärung. Jeder nimmt täglich Services in Anspruch. Es handelt sich um eine Leistung, die ein Dritter für uns erbringt, von der wir vor der Inanspruchnahme wissen, was sie uns bringt, was sie kostet und in welchem Zeitraum sie erbracht wird. Wie gut oder schlecht diese Leistung letztendlich durchgeführt wird und welche Konsequenzen daraus entstehen, ist ein Thema für sich, was mehr als Technik beinhaltet.

In Abbildung 1 sind die Elemente einer Service-orientierten Architektur, dem „SOA-Tempel“, wiedergegeben. Basis der gesamten Architektur sind die drei Stufen Akzeptanz, Sicherheit und Einfachheit. Alle diese Eigenschaften erscheinen mittlerweile so selbstverständlich, dass sie gerne als gegeben angesehen werden. Auf diesem Fundament ruhen die vier Säulen (Merkmale), die zusammen eine Service-orientierte Architektur stützen.

▼ **Verteilung:** Mit der stetig voranschreitenden Vernetzung unseres Lebens und der Unternehmen ist Verteilung ein Aspekt, den jede moderne Architektur unterstützen muss. Inseldasein ist nur noch in Ausnahmefällen sinnvoll. Damit einhergehend ist die Forderung nach einem systemneutralen Zugang zu einem Service, d. h. der Service-Anbieter sollte lediglich einen minimalen Satz technischer Parameter festlegen.



▼ **Lose Kopplung:** Zu der voranschreitenden Vernetzung gesellt sich die immerschnellere Änderung der Umgebung. Auf technischem Gebiet steigt das Tempo der Innovationszyklen und im wirtschaftlichen Umfeld werden permanent neue Partnerschaften eingegangen und wieder gelöst. Das heißt, die Interna ändern sich, aber die Dienstleistungen (Services), die ein Unternehmen anbietet, bleiben (nahezu) gleich. Der Gewinn aus der losen Kopplung ist eine Verringerung des Anpassungsrisikos der Dienstanutzer aufgrund externer Änderungen.

▼ **Standards:** Die Verwendung von Standards bietet zwei wichtige Vorteile. Erstens, können alle Beteiligten in einem Service-Szenario die Implementierung einer Laufzeitumgebung von entsprechend spezialisierten Unternehmen oder Organisationen erwerben und sich auf das oft apostrophierte „Kerngeschäft“ konzentrieren. Zudem bietet die Verwendung ein gewisses Maß an Investitionssicherheit, da im Zweifelsfall auf andere Anbieter ausgewichen werden kann.

▼ **Prozessorientiert:** Eine Service-orientierte Architektur gibt zwar keine Beschränkungen bzgl. der Komplexität eines Services vor, dennoch sollten Service-Implementierungen mehr auf „kleine“ (elementare) Leistungen fokussieren. Dem Service-Anbieter bietet dieses Vorgehen den Vorteil, in kürzerer Zeit und mit geringerem Risiko einen Service zu entwickeln und anbieten zu können. Für den Service-Nutzer steht ein Service schneller zur Verfügung. Komplexe Services werden nicht implementiert, sondern zusammengesetzt aus einer Abfolge von Teil-Services. Der Ablauf der Teil-Services (Prozess) wird deklarativ definiert, wofür sich die Begriffe „orchestriert“ oder „choreographiert“ eingebürgert haben.

Bleibt abschließend noch zu klären, wer die Nutzer (Akteure) einer Service-orientierten Architektur sind. Im Unterschied zu einer Browser-Anwendung steht im SOA-Konzept die automatisierte Kommunikation zwischen

Anwendungen im Mittelpunkt der technologischen Überlegungen. Letztendlich ist zwar der Mensch der Nutznießer, im SOA-Konzept allerdings „nur“ mittelbar.

## Web Services

In Abbildung 2 ist die wohlbekannte Dreiecksbeziehung mit den Akteuren Requestor, Provider und Broker einer Service-orientierten Architektur dargestellt. Üblicherweise wird diese Darstellung sogleich mit den für Web Services spezifischen Komponenten SOAP, WSDL und UDDI dargestellt, was mitunter suggeriert, dass Service-orientierte Architektur und Web Services lediglich Synonyme sind. Um das volle Potenzial einer SOA entfalten zu können, ist es jedoch wichtig, Web Services als eine von mehreren Implementierungsmöglichkeiten des Architekturansatzes zu erkennen. Andere, gleichwertige Möglichkeiten Service-orientierte Architekturen umzusetzen sind CORBA, DCOM und RMI.

Ein Vorteil von Web Services ist es, dass sie auf den Standards der XML-Sprachfamilie basieren, was zusammen mit der großen Akzeptanz dieser Technik auch die breite praktische Umsetzung von Web Services gefördert hat. Da in den Anfangstagen dieser Technik das Web aus „Web Services“ häufig mit HTTP gleichgesetzt wurde, konnte zumindest eine einfache basale Absicherung des Datenaustausches per SSL/TLS erreicht werden.

Ein Vorteil der Nutzung von HTTP in Web Services ist, dass sie damit per se geeignet sind zur Entwicklung von verteilten Anwendungen. Die anfängliche Fokussierung auf das RPC-artige\*\* Kommunikationsmuster ermöglicht es zudem vergleichsweise schnell und unkompliziert durch Codegeneratoren notwendige Schnittstellen aufzubauen. Den Aspekt der Einfachheit können Web Services erfüllen, da im Rahmen einer Web Services-Kommunikation „lediglich“ Daten kopiert werden. Im Gegensatz zu anderen Konzepten – z. B. CORBA – werden Service-Nutzer und

Anbieter damit nicht über eine gemeinsame Objektnetzstruktur miteinander gekoppelt.

Die großen Herausforderungen für die Web Services-Technologie resultieren aus den Anforderungen der Prozessorientierung. Diese Anforderung stellt die bisher üblichen – jedoch nicht von der Spezifikation erzwungenen – Annahmen über einige Aspekte von Web Services in Frage. Zur Verdeutlichung wird diese Rückkopplung am Beispiel von SOAP kurz erläutert:

▼ **Kommunikationsstil:** Der bevorzugte Kommunikationsstil bei Web Services war bisher der RPC-Stil. Dadurch wird jedoch eine Kopplung erzwungen, die der benötigten Flexibilität für eine Prozessorientierung im Wege stehen kann. Eine Alternative ist die asynchrone nachrichten-orientierte Kommunikation.

▼ **Transportprotokoll:** Die Kommunikation mit Hilfe des Transportprotokolls HTTP erweist sich aufgrund des niedrigen Datendurchsatz und der Unzuverlässigkeit zunehmend als ein Hindernis, Web Services in allen Netzen (Inter, Extra, Intra) einzusetzen. Mittlerweile existieren SOAP-Laufzeitumgebungen verschiedenster Hersteller, die neben HTTP auch JMS, RMI oder die direkte Verwendung von TCP als Transportprotokoll anbieten.

▼ **Sicherheit:** Wie oben bereits beschrieben, ist eine wichtige Grundlage der Prozessorientierung, dass mehrere Services verschiedener Anbieter zu einem neuen größeren Dienst zusammengesetzt werden können. Damit müssen auch Fragen nach Verbindlichkeit, partieller Verschlüsselung einer Nachricht und Sicherheitskontext beantwortet werden. Der einfache Ansatz HTTP über SSL/TLS ist hierbei nicht mehr ausreichend.

## Automatisierte Dienstnutzung

Der einfache Teil der oben beschriebenen Anforderungen der Prozessorientierung ist, dass

\*\* Damit ist nicht notwendigerweise, die in der SOAP-Spezifikation beschriebene Syntax gemeint bzw. notwendig.

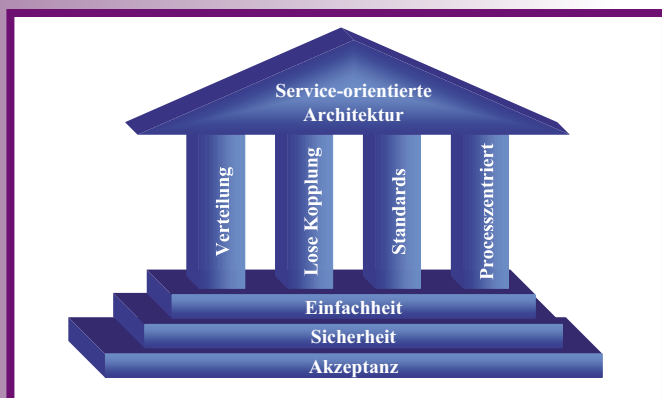


Abb. 1: Der Tempel der Service-orientierten Architektur, angelehnt an [ST]

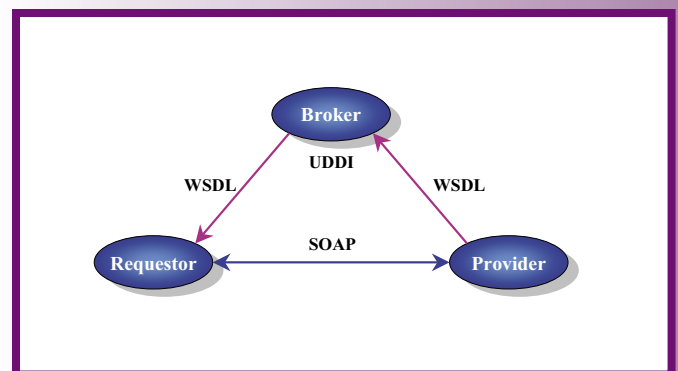


Abb. 2: Das Dreieck einer Service-orientierten Architektur angereichert mit den grundlegenden Web Services-Komponenten



in den Bereichen Kommunikation, Transport und Sicherheit „lediglich“ eine Überarbeitung und Erweiterung bestehender Konzepte benötigt wird. Komplizierter gestaltet sich der Sachverhalt bei der Automatisierung der Dienstnutzung und damit der Prozessoptimierung. Woher weiß eine Anwendung, welche Services sie nutzen soll, um ein gesetztes Ziel zu erreichen? Bei genauer Analyse dieser Frage rückt der Aspekt der Bedeutung (Semantik) eines Services in den Mittelpunkt.

Betrachtet man die bisherige Entwicklung der Internet-Technologie, so fällt auf, dass die bis dato verfügbaren Spezifikationen hauptsächlich einen syntaktischen Rahmen definieren. Bei der Beschreibung eines Services mittels Web Service Description Language (WSDL) verbleibt die Bedeutung des Services im Kopf des Anbieters und wird nicht in das WSDL-Dokument übertragen; allenfalls gibt es nicht maschinenverwertbare Kommentare. Diese fehlende explizite Beschreibung der Semantik wird zunehmend als Hindernis für eine „vollständige“ Automatisierung eingestuft [DB].

Die Ursachen dafür, dass Web Services hauptsächlich auf syntaktischen Spezifikationen basieren, ist in ihrer Historie begründet. In Abbildung 3 ist dargestellt, dass Web Services ihren Ursprung im World Wide Web (WWW) haben. Das WWW mit Browsern als clientseitigen Endgerät ist auf die direkte Interaktion eines vernunftbegabten menschlichen Akteurs mit einer Anwendung ausgerichtet. Die Bedeutung einer Web-Seite entsteht im Kopf des Akteurs. Die Seitenbeschreibungssprache HTML muss lediglich Daten und Angaben über deren Präsentation übermitteln. Dazu muss der Browser nicht „wissen“ bzw. ableiten können, was die dargestellte Information bedeutet. Syntaktische Angaben, wie „dies ist eine Überschrift erster Ebene“ (h1), reichen völlig aus.

Web Services stellen einen wichtigen Schritt in Richtung Automatisierung des Datenaustauschs dar. Sie geben den syntaktischen Rahmen vor, wie Services beschrieben (WSDL), gefunden (UDDI) und Daten ausgetauscht werden (SOAP). Bis auf die Auswahl des geeigneten Dienstes kann somit ein großer Teil der Kommunikation bereits weitgehend automatisiert ablaufen.

Der Bedarf nach Semantik zeichnete sich jedoch bereits in den Anfängen der Web Services ab. Interessanter Weise manifestierte sich dies am deutlichsten in der Komponente UDDI, die aktuell etwas in das Kreuzfeuer der Kritik geraten ist. Momentan ist unklar, wie diese Komponente sinnvoll genutzt werden kann. Bereits jetzt können Service-Anbieter mit Hilfe von Klassifizierungssystemen (Taxonomien) wie DUNAS\*\*\* eindeutig eingeordnet werden. Das heißt, der Eintrag in der UDDI ist mit einer bestimmten Bedeutung verbunden.

Das Auffinden eines bestimmten Services ist

allerdings lediglich über eine bloße Schlüsselwortsuche realisiert. Für eine Automatisierung ist dies i. d. R. nicht ausreichend, da die Begriffe des Alltags nicht immer eindeutig bzw. kontextabhängig sind. Deshalb bieten neuere UDDI-Implementierungen die Möglichkeit auch Services explizit mit Bedeutung zu versehen.

Eine vollständig automatisierte Nutzung von Web Services wird jedoch erst dann möglich sein, wenn alle Komponenten in der Lage sind, Semantik in einer maschinenlesbaren Form abzulegen und zugreifbar zu machen und gegebenenfalls auswerten können. Bei der Erweiterung von Web Services um Semantik verfolgt die Web Services-Gemeinde den bisher erfolgreichen Weg, bereits existierende Standards, die eine gewünschte Funktionalität bieten, zu integrieren. Im Fall von Semantik ist dies das Semantic Web.

### Semantic Web

Im Grunde fasst das Schlagwort „Semantic Web“ seit 2001 alle zuvor isoliert vorangetriebenen Bestrebungen, das Web in seiner gegenwärtig von HTML-Inhalten dominierten Form um „Bedeutung“ anzureichern, zusammen. Ausgangspunkt der Begriffsbildung ist ein von Tim Berners-Lee, James Hendler und Ora Lassila verfasster Artikel [BLH+01]. Darin stellen die Autoren ihre Vision einer maschinengestützten Interaktion zwischen Menschen vor, die über die gegenwärtig anzutreffenden technischen Unterstützungsformen wie E-Mail, Web-Seiten und mobile Telephonie hinausreicht.

Die hierzu verwendete Technik unterscheidet sich von den aktuell gängigen „elektronischen Helferlein“ dadurch, dass die zur Aufgabenerfüllung heranzuziehenden Daten (wie bereits vereinbarte Termine, freie Zeit und persönliche Planungspräferenzen) in ihrer Bedeutung der Maschine „bekannt“ sind. Die notwendige Bedeutungszuschreibung erfolgt hierbei ausschließlich durch den Menschen, weshalb ein Kalender auch zukünftig nichts über eingetragene Geburtstage „wissen“ muss, sondern lediglich anhand des Eintragstyps und seiner festgelegten Charakteristika erkennt, dass es sich um einen nicht verschiebbaren Termin handelt.

Diese weiterhin bestehende tragende Rolle des Menschen ist daher gleich in zweifacher Hinsicht hervorzuheben. Das semantische Web trachtet nicht danach die Aufgaben von Menschen zu übernehmen oder diesen zu ersetzen, den beteiligten Maschinen wird nicht einmal die Intelligenz zugebilligt, selbst automatisiert Schlüsse aus den durch sie verwalteten Daten zu ziehen. All diese Visionen, die an den altbekannten HAL aus „2001“ oder den etwas vorlauten „C3-PO“ erinnern, werden weiterhin im Reich der „künstlichen Intelligenz“ (KI) belassen, deren Verschmelzung mit dem Web ausdrücklich nicht Zielsetzung des Semantic Webs ist. Vielmehr versteht sich diese Vision als Komplementierung des bestehenden Webs um Aspekte der Maschine-Maschine-Kommunikation; wenngleich auch einige Termini und Basistechniken dieses Forschungszweiges Eingang in die Aktivitäten gefunden haben.

Vor diesem Hintergrund wird auch offenbar, dass diese Erweiterung des heutigen Webs keines „Semantic Web Browsers“ als Präsentationskomponente bedarf, da die erzeugten, ausgetauschten und verarbeiteten Daten ausschließlich der Konsumption durch maschinelle Agenten vorbehalten sind. Als „Agent“ wird hierbei jede Softwarekomponente aufgefasst, die im Namen einer natürlichen oder juristischen Person oder eines Prozesses agiert. Diese pragmatische Definition grenzt absichtsvoll weitergehende Ansätze (wie autonome Auftragserfüllung oder dynamische Zustandsmigration) der KI aus. Zudem wird gleichzeitig die begriffliche Kompatibilität zum klassischen WWW gewährleistet, welches durch „Web Agenten“ (d. h. Browser) bedient wird.

\*\*\* Mit Hilfe dieser neunstelligen Nummer kann ein Unternehmen eindeutig identifiziert werden. Die Nummern werden zentral von einem Unternehmen verwaltet.

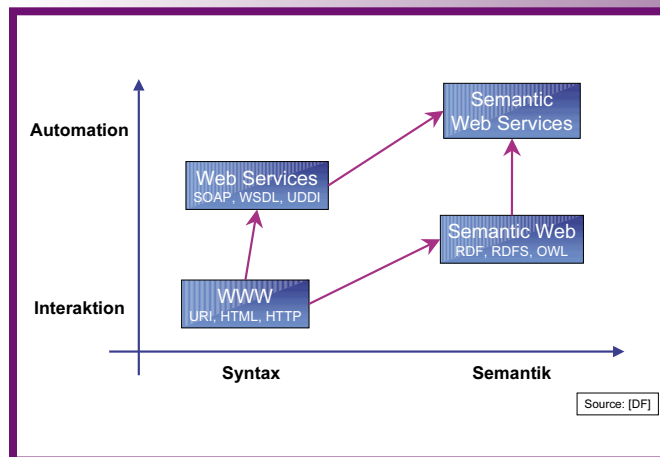


Abb. 3: Einordnung der Semantik-Web Services-Konstituenten bzgl. Automation und Semantik, angelehnt an [DF]



Gleichzeitig wird das semantische Web als Erweiterung der beiden Basiselemente „Hypertext-Seite“ und „Link“ des sichtbaren HTML-basierten verstanden. Während die herkömmlichen Hyperlinks des Ankerelements es bereits gestatten, innerhalb einer HTML-Seite auf Inhalte beliebigen Typs – etwa andere Web-Seiten, Dokumente oder Binärdaten – zu verweisen, fasst das semantische Web die Beziehungsquellen und -ziele einheitlich als „Ressourcen“ auf. Zwischen den, anhand der Linkstruktur zugänglichen, Ressourcen werden jedoch nicht mehr ausschließlich gleichartige Beziehungen etabliert, sondern die gebildeten Verknüpfungen anhand ihrer Bedeutung unterschieden. So kann eine Ressource durch eine Beziehung des Typs „erstellt durch“ auf eine andere Ressource verweisen, die Daten über den Autor bereithält, während diese wiederum durch einen Link des Typs „E-Mail-adresse“ auf eine Ressource verweist, welche die Mailadresse des Autors enthält.

Bereits dieses Beispiel lässt deutlich werden, dass die denkbaren Beziehungstypen so zahlreich sind, dass keine zentrale Instanz diese vorhersehen oder vorgeben kann. Auch mit dieser Erkenntnis stützt sich das semantische Web auf den Grundideen des WWW ab. Seinem Ansatz verteilter Datenablage liegt die Überzeugung zu Grunde durch Dezentralisierung der Verantwortung ein größeres System

stärker funktionsfähig zu halten (relativ zur Gesamtgröße gesehen sind Fehler der Nummer 404 vergleichsweise selten, wenngleich selten ärgerlich), als es eine Umsetzung mit genau einer Koordinierungsinstanz erlauben würde. Daher steht es dem Autor im semantischen Web offen, die von ihm benötigten Beziehungstypen auf geeignete Art und Weise selber zu definieren. Die Technologie, mit der diese Bedeutungszuweisung vorgenommen wird, sollte im Sinne einer SOA auf Standards beruhen. Zurzeit wird die XML-Anwendung Resource Description Framework (RDF) favorisiert, die wir im nächsten Artikel eingehender betrachten werden.

### Zusammenfassung

Ziel dieses Artikels war es den Bedarf von Semantik innerhalb einer Service-orientierten Architektur herauszuarbeiten, da ohne diese die Dreiecksbeziehung der Web Services zum „Bermudadreick“ geraten würde, in dem sich Unglücke während des Dienstaufrufs „unerklärlich“ häufen würden. Zur Beleuchtung der Hintergründe wurden zuvorderst die Anforderungen an eine SOA beschrieben. Anschließend wurde am Beispiel von Web Services – der aktuell populärsten SOA-Inkarnation – dargestellt, dass insbesondere

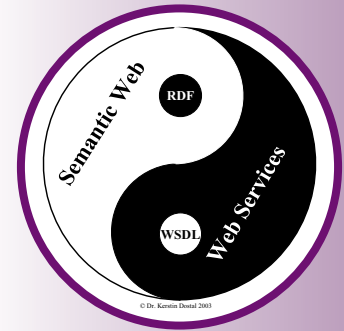


Abb. 4: Semantische Web Services als Symbiose von Semantik (Yin) und Syntax (Yang)

die Anforderung der Prozessorientierung mit der automatisierten Service-Auswahl nur mit Semantik vollständig erfüllt werden kann.

Eine Service-orientierte Architektur kann demzufolge nur entstehen, wenn Syntax (Web Services) und Semantik (Semantic Web) ein gemeinsames Ganzes bilden. Dies kommt unserer Ansicht nach sehr gut im Yin-Yang-Bild zum Ausdruck. Wie in Abbildung 4 dargestellt, ist die Kerntechnik des Semantic Webs das RDF-Format. Dieses und ergänzende Techniken werden wir im nächsten Artikel vorstellen und beschreiben, wie XML-Dokumente mit Semantik angereichert werden können.



**Dr. Wolfgang Dostal** ist für die IBM Global Services (AMS SI) als IT-Architekt tätig. Er befasst sich intensiv mit Internettechnologien und Anwendungsintegration (EAI). Der Focus seiner Arbeit liegt auf der Erstellung von J2EE-Anwendungen und der Einführung von Web Services. An der Fachhochschule Mainz unterrichtet er zum Thema objektorientierte Softwareentwicklung. E-Mail: wolfgang.dostal@de.ibm.com.

**Mario Jeckle** hat eine Professur für Software Engineering an der Fachhochschule Furtwangen inne und arbeitet schwerpunktmäßig im Umfeld XML und Datenmodellierung. Als Mitglied des Advisory-Committees vertritt er die DaimlerChrysler-Forschung im W3C. E-Mail: mario@jeckle.de.

### Literatur und Links

**[BLH+01]** T. Berners-Lee, J. Handler, O. Lasila, The Semantic Web, Scientific American, Mai 2001

**[DB]** D. K. Barry, Web Services and Service-oriented Architectures, Morgan Kaufmann Publishers, ISBN 1-55860-906-7

**[DF]** D. Fensel, Next Web Generation, Leopold-Franzens-Universität Innsbruck

**[GAR]** Gartner Research Note T-17-5338, 20. August 2002

**[ST]** An Analysis Memo from The Stencil Group, April 2002, The Laws of Evolution: A Pragmatic Analysis of the Emerging Web Services Market

**[WA]** Web Services Architecture, <http://www.w3.org/TR/2003/WD-ws-arch-20030808/>