



Rettung für Babylon

Semantik und Web Services: Vokabulare und Ontologien

Wolfgang Dostal, Mario Jeckle, Werner Kriechbaum

Bedeutung zu transportieren ist das grundlegende Ziel einer Kommunikation. Dabei hat jeder schon erleben müssen, dass die eigenen gedanklichen Verknüpfungen zwischen den verwendeten Worten und den damit einhergehenden Vorstellungen beim Kommunikationspartner nicht notwendigerweise reproduzierbar sind. Daher entsteht häufig ein nicht unerheblicher Aufwand, um die Vokabulare und die dahinterstehende Bedeutung gegeneinander abzugleichen. In Alltagssituationen, wie dem Überqueren einer Ampelkreuzung oder vergleichbar stark reglementierten Vorgängen, besteht kaum noch Klärungsbedarf über die Bedeutung der Zeichen bzw. Vokabeln. Das heißt jedoch nicht, dass es keinen Aufwand für den Abgleich gab. Vielmehr wurde er von den meisten schon in der Kindheit erbracht.

► Im vorangegangenen zweiten Artikel dieser Serie [DoJeKr04] haben wir beschrieben, wie mit Hilfe des Resource Description Frameworks (RDF) eine Maschinen-auswertbare Semantikdarstellung erstellt werden kann. Dazu stellt dessen Spezifikation einen überschaubaren abstrakten Satz von Vokabeln zur Verfügung. In der Regel ist dieses jedoch nicht ausreichend, um die Sachverhalte eines Anwendungsgebiets (Domäne) hinreichend aussagekräftig zu beschreiben. Deshalb stellt RDF mit RDF-Schema (RDFS) ein zusätzliches Konzept zur Verfügung, welches es erlaubt eigene Vokabulare zu definieren und diese wiederum mittels RDF auszudrücken.

Allerdings zeigt die Erfahrung, dass es organisatorisch und politisch nahezu aussichtslos ist eine Übereinkunft hinsichtlich eines global gültigen Vokabulars zu etablieren, auf das alle Kommunikationspartner zurückgreifen. Um die Anforderung der losen Kopplung aus der Service-orientierten Architektur (SOA) (s. 1. Teil, [DoJe04]) nicht aufgeben zu müssen, wird daher ein Konzept benötigt, mit dem es möglich ist, eigenständig lokal definierte Vokabulare gegeneinander abgleichen zu können. Im Rahmen dieses Artikels werden wir den W3C-Ansatz der Web-Ontologien (OWL) vorstellen, welche sich zum Ziel setzen einen Beitrag zur Lösung dieser Heraus-

schiedener Vokabulare aufeinander abzubilden sind.

Beginnend wird aufgezeigt, welche Schwierigkeiten zu Tage treten können, wenn verschiedene Domänen eigene als Vokabulare dienende RDF-Instanzen definieren. Die zentrale Forderung der syntaktischen Uniformität hinsichtlich der Beschreibungssyntax von Identifikatoren auf Basis der Uniform Resource Identifier (URI) haben wir in diesem Zusammenhang bereits im vorangegangenen Artikel dargestellt.

Identifikatoren und RDF

Eine zentrale Aufgabe der Domänenexperten ist die Bildung von eindeutigen Identifikatoren innerhalb eines Vokabulars. Unabhängig davon, ob natürlichsprachliche Worte oder syntaktisch formalisierte URIs zum Verweis auf Begriffe – also Konzepte, die jenseits der lautgestaltlichen Repräsentation existieren – verwendet werden, müssen diese idealerweise eineindeutig sein. Erst die Erfüllung dieser Forderung, welche dem Wunsche nach der Verfügbarkeit genau einer Benennung für einen abstrakten Sachverhalt und gleichzeitig dem Angebot genau eines

identifizierenden Wortes für jeden Sachverhalt gleichkommt, bildet die Voraussetzung austauschbarer Semantikbeschreibungen.

Angewandt auf die Bereitstellung einer korrekt in eindeutiger Weise durch eine Maschine verwertbaren Beschreibung semantischer Eigenschaften eines Web Service bedeutet dies, dass alle Kommunikationspartner nicht nur dieselbe Syntax (etwa RDF/XML) verwenden müssen, sondern zusätzlich, dass die Beschreibung identischer Konzepte mittels derselben Worte (bzw. Identifikator) erfolgen muss. Übertragen auf das alltägliche Leben würde dies bedeuten, dass alle Beteiligten dieselbe Sprache verwenden und durch dasselbe Alphabet und dieselbe Grammatik ihre Vorstellungen formulieren. Darüber hinaus müsste gefordert werden, dass alle Kommunizierenden mit einem Wort (Begriff) dasselbe Konzept verbinden.

Angesichts dieser Forderung für mehr als einen einzigen Kommunikationsteilnehmer kaum erfüllt werden. Und auch in Selbstgesprächen entsteht – wie in [Tol03] mit der Figur des *Gollum* eindrucksvoll gezeigt – mitunter der Bedarf nach Klärung der genauen Begriffsbedeutung*. Daher unterneh-

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:sws="urn:example.com/sws#">
  <rdf:Description rdf:about="urn:uddi-org:api_v3#businessService">
    <rdf:type rdf:resource="urn:myServices#Rechnung"/>
    <sws:WSDLRef>http://www.example.com/9437.wsdl</sws:WSDLRef>
    <sws:Addition>...</sws:Addition>
    <sws:Substraktion>...</sws:Substraktion>
    <sws:Multiplikation>...</sws:Multiplikation>
    <sws:Division>...</sws:Division>
  </rdf:Description>
</rdf:RDF>
```

Listing 1: RDF/XML-Beschreibung eines Web-Dienstes

* Obwohl in diesen Zusammenhang trefflich darüber philosophiert werden kann, ob es sich nicht doch um ein Zwiegespräch (Gollum vs. Smeagol) handelt.



```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ser="urn:example.org/ser#"
  xmlns:x="urn:myServices#">
  <x:Calculator rdf:about="urn:uddi-org:api_v3#businessService">
    <ser:InterfaceRef rdf:parseType="Resource">
      <rdf:value>http://www.example.com/9437.wsdl</rdf:value>
    </ser:InterfaceRef>
    <ser:ArithGrFkt>...</ser:ArithGrFkt>
  </x:Calculator>
</rdf:RDF>

```

Listing 2: Semantisch gleichartige Beschreibung eines gleichwertigen Web-Dienstes

men weder RDF noch die darauf aufbauenden Standards den Versuch, die Beschreibungsmöglichkeiten dergestalt zu reglementieren. Vielmehr wird jeder Anwender ausdrücklich dazu ermutigt die Begrifflichkeit seiner Domäne beizubehalten und zur Beschreibung der gewünschten Semantikzusammenhänge zu verwenden. Allerdings ist auf dieser Basis die Eindeutig- und Einheitlichkeit der entstehenden Beschreibungen zwischen den Domänen – und auch von Gruppen innerhalb einer Domäne – nicht mehr sichergestellt. So können verschiedene RDF-Instanzen dieselben Sachverhalte in unterschiedlicher Weise, d. h. durch die Wahl verschiedener Identifikatoren, ausdrücken.

Die Gegenüberstellung der RDF-Instanzen aus Listing 1 und 2 zeigt drei der auftretenden Probleme bei der Semantikbeschreibung anhand eines einfachen Dienstes. Der Dienst unseres Beispiels stellt die vier einfachen arithmetischen Grundrechenoperationen zur Verfügung und expliziert die angebotenen Schnittstellen durch eine WSDL-Beschreibung, die in der RDF-Beschreibung referenziert wird.

Auf den ersten Blick fallen die strukturellen Unterschiede der semantisch identischen RDF-Aussagen ins Auge. So wählen beide Dienste verschiedene Elementnamen für das Element, welches das `rdf:about`-Attribut enthält, referenzieren dieselben WSDL-Schnittstellenbeschreibungen und bieten mehrere Dienste (**Addition**, **Subtraktion**, **Multiplikation**, **Division** bzw. **ArithGrFkt**) an.

Allgemein gefasst gehören diese Unterschiede drei Klassen an [Sei80]:

- ▼ **Übersetzungen:** Offenkundig handelt es sich bei den gewählten Typidentifikatoren `urn:myServices#Rechnung` und `urn:myServices#Calculator` „nur“ um Übersetzungen. Allerdings sind weder die verwendeten Sprachen offensichtlich (so kann es sich beim *Calculator* gleichermaßen um das englische Wort für *Rechner* oder das lateinische für *Rechenmeister* handeln) noch die Annahme naheliegend, ein maschineller Interpret könne diese leisten.
- ▼ **Synonyma:** Unterstellt man, dass beide Dienste dieselbe Leistung anhand identischer Schnittstellen (kenntlich an der Verwendung derselben URI) anbieten, so

kann es sich bei den beiden referenzierten WSDL-Beschreibungen (Element `sws:WSDLRef` bzw. `ser:InterfaceRef`) nur um verschiedene Namen für denselben Sachverhalt handeln.

▼ **Beide Definitionsseiten:** Offenkundig benennen die Elemente **ArithGrFkt** und **Addition**, ... dieselbe Bedeutung, wobei jedoch in Listing 1 die ausführ-

liche inhaltliche Definition gegeben ist und in Listing 2 die kurze fachsprachliche Bezeichnung Verwendung findet.

Wie die Beispiele in Listing 1 und 2 zeigen, kann RDF alleine nur einen Teil der Herausforderungen, die durch das Bereitstellen Maschinen-lesbarer Semantikbeschreibungen auftreten, bewältigen. Im Kern weist das zu lösende Problem zwei Aspekte auf:

▼ **Struktursyntaktische Unterschiede:** Die XML-Darstellung von RDF beschreibt Aussagen über den einen Dienst aus Sicht eines Autors. Die Beschreibung desselben Dienstes durch einen anderen Autor kann – selbst unter Verwendung von RDF – strukturell und/oder inhaltlich von dieser abweichen, auch wenn dieselben semantischen Aussagen getroffen werden.

▼ **Verwendung von unterschiedlichen Namen:** Selbst bei identischer syntaktischer Struktur der Beschreibungen können immer noch unterschiedliche Namen (Identifikatoren) für denselben realen Sachverhalt verwendet werden.

RDF-Schema

RDF selbst ist, wie im zweiten Teil der Serie vorgestellt, lediglich ein Formalismus zur Repräsentation von Informationsbeschreibungen. Dieser ist – anders als viele Spezifikationen des W3C – nicht an eine bestimmte lexikalische Repräsentation gebunden, sondern kann durch verschiedene Serialisierungsmechanismen ausgedrückt werden. In Listing 1 und 2 wurde das RDF/XML-Format gewählt (d. h. die Darstellung von RDF als XML), da diese Darstellung sich vergleichsweise harmonisch in die ebenfalls XML-basierten Web Services einfügen lässt.

Aufgrund der durch RDF bewusst nicht festgelegten syntaktischen Darstellung können im WWW eingeführte Grammatiksprachen zur Struktur- und Inhaltsbeschreibung, wie die dem klassischen HTML zugrunde liegenden Dokumenttypdefinitionen (DTD) oder die im XML-Umfeld gebräuchlichen XML-Schemata, nicht eingesetzt werden, da diese jeweils genau eine festgelegte Syntax zur Darstellung der

Grammatikinstanzen vorsehen. Überdies sind sowohl DTDs als auch XML-Schemata einerseits zu restriktiv, andererseits in den formulierbaren Aussagen zu schwach, um RDF-Strukturen ausdrücken zu können.

So benötigen RDF-Beschreibungen, anders als Datenbeschreibungen, kein Typsystem und sehen daher keine Möglichkeit zur Restriktion eines auftretenden Inhalts vor. Zwar können Interpretationshinweise innerhalb des RDF/XML-Formates hinterlegt werden, die sich des Typsystems aus XML-Schema Teil 2 bedienen, diese sind jedoch nur dokumentarisch und werden nicht zur Validierung des Dokuments herangezogen. Ebenso verhält es sich mit dem in den XML-Grammatiksprachen anzutreffenden Zwang zur Ausdetaillierung der syntaktischen Strukturrepräsentation, etwa durch Festlegung der Auftretensreihenfolge und Häufigkeit von Elementen sowie die Entscheidung für die Darstellung eines gegebenen Datums als Attribut oder Element. Für die Repräsentation ist diese Konkretisierung zwar essentiell, für die dargestellte Bedeutung ist dies jedoch unerheblich.

Gleichzeitig reicht der durch die angebotenen XML-Grammatiksprachen vorgesehene Umfang zur Beschreibung von RDF an vielen Stellen nicht aus. Während DTD und XML-Schema lediglich (semantikfreie) Attribut-einbettungen und Kindelementbeziehungen vorsehen, verfügt RDF über eine Reihe spezialisierter Beziehungen zwischen den einzelnen Sprachelementen. In diesem Sinne bezeichnet das `rdf:about`-Attribut immer das Subjekt einer Aussage, bzw. deren Identifikator. Hierbei handelt es sich nicht nur um ein Attribut des Namens `about` im RDF-Namensraum, sondern um ein Sprachelement der Metaebene, dessen Bedeutung und lexikalische Benennung außerhalb der verwendenden RDF-Sprache und ebenso unabhängig von deren syntaktischer Repräsentation festgelegt ist.

Aus den geschilderten Gründen existiert für RDF keine normative DTD oder eine XML-Schemabeschreibung – da diese beiden Grammatikmechanismen andere Ziele (konkret: strukturelle und inhaltliche Validierung) verfolgen. Vielmehr wird zur Beschreibung der Bedeutungsstrukturen von RDF auf RDF selbst zurückgegriffen. Die so unter der Bezeichnung *RDF-Schema* [RDFS] standardisierte RDF-basierte Beschreibungssprache für RDF-Strukturen gestattet es, die erwarteten Inhalte einer RDF-Beschreibung semantisch zu spezifizieren, ohne Aussagen über deren lexikalische Darstellung treffen zu müssen.

Listing 3 zeigt einen Ausschnitt der RDF-Schemabeschreibung des in Listing 2 verwendeten RDF-Vokabulars. Das Schema definiert zunächst den Namensraum (`urn:uddi-org:api_v3#`), in dem sich alle selbsterstellten

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  <rdf:Description rdf:about="urn:uddi-org:api_v3#"
  <rdf:comment>Namensraum zur Beschreibung von Web-Diensten
  </rdf:comment>

  <rdf:seeAlso rdf:resource="http://www.jeckle.de/semanticWebServices/" />
  </rdf:Description>
  <rdf:Class rdf:about="urn:uddi-org:api_v3#businessService"
  <rdf:isDefinedBy rdf:resource="urn:uddi-org:api_v3#" />
  <rdf:comment>Der Service</rdf:comment>
  </rdf:Class>
  <rdf:Property rdf:about="urn:example.org/ser#InterfaceRef"
  <rdf:isDefinedBy rdf:resource="urn:example.org/ser#" />
  <rdf:label>InterfaceRef</rdf:label>
  <rdf:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI" />
  <rdf:domain rdf:resource="urn:uddi-org:api_v3#businessService" />
  </rdf:Property>
  <rdf:Property rdf:about="urn:example.org/ser#ArithGrFkt"
  <rdf:isDefinedBy rdf:resource="urn:example.org/ser#" />
  <rdf:label>ArithGrFkt</rdf:label>
  <rdf:range rdf:resource="
  "http://www.w3.org/2000/01/rdf-schema#Literal" />
  <rdf:domain rdf:resource="urn:uddi-org:api_v3#businessService" />
  </rdf:Property>
  </rdf:RDF>
```

Listing 3: RDF-Schema zur Beschreibung des Vokabulars für unser Beispiel aus Listing 2

(d. h. nicht im RDF-Standard vorgesehenen) Aussagen befinden. Daran schließt sich mit **Class** die Definition einer Ressource an, die durch eine URI identifiziert wird. Zusätzlich sollte für jedes als Ressource dargestellte Subjekt die Zuordnung zum beherbergenden Namensraum durch das **isDefinedBy**-Prädikat angegeben werden.

Die in der RDF-Serialisierung zu einer Ressource zuordenbaren Prädikate werden als **Property**-Elemente dargestellt. Auch sie werden (durch das **rdf:about**-Attribut) durch eine URI identifiziert und durch den im **label**-Element angegebenen Namen benannt. Dieser wird in der späteren Verwendung in einer RDF/XML-Instanz als Element- oder Attributname innerhalb desjenigen Namensraumes erwartet, der durch das **isDefinedBy**-Attribut festgelegt wird. Die Bindung eines Prädikats an ein definiertes Objekt erfolgt durch das **rdfs:domain**-Element, konkret die URI-Referenz in dessen **rdf:resource**-Attribut.

Zusätzlich kann für jedes Prädikat ein Hinweis auf den in der RDF-Instanz erwarteten Datentyp gegeben werden. Dieser wird als URI im **range**-Attribut abgelegt. Hierbei illustriert Listing 3 eine Besonderheit von RDF-Schema: Es gestattet die Verwendung (durch URI-basierte Referenzierung) beliebiger Typsysteme neben den in RDFS vorgesehenen basalen Typen. So wird den später durch das Prädikat **InterfaceRef** zugänglichen Werten der Typ **anyURI** aus XML-Schema Teil 2 zugewiesen, während die **ArithGrFkt** als **Literal** typisiert beliebige Kombinationen aus Freitexten und/oder Zahlen aufnehmen kann.

Zusammenfassend gilt folgende Gegenüberstellung zwischen RDF-Grundprimitive

und RDFS-Beschreibung:

- ▼ Subjekt/Ressource: **rdfs:Class**,
 - ▼ Prädikat/Eigenschaft: **rdfs:Property**,
 - ▼ Objekt/Wert: typisiert durch **rdfs:range** und Inhaltsangabe in RDF-Instanz.
- Auf diesem Wege schafft RDFS die Voraussetzung zur syntax- und serialisierungsneutralen Beschreibung der Struktur- und Inhaltscharakteristika eines RDF-basierten Vokabulars und ermöglicht es so Beschreibungstypen zu erstellen, welche gleichartige Aussagen über Ressourcen repräsentieren.

Allerdings löst RDF-Schema nur den strukturellen Teil der Herausforderung.

Steht dem Beschreibenden eines Web-Dienstes ein bestehendes RDF-Schema nicht zur Verfügung oder besitzt er schlichtweg keine Kenntnis von dessen Existenz, so erzeugt er möglicherweise ein neues, welches eine vom bestehenden abweichende Terminologie einführt. So wird der in Listing 1 als **WSDLRef** benannte Verweis auf eine in WSDL formulierte Schnittstellenbeschreibung in Listing 2 als **InterfaceRef** bezeichnet.

Ontologien

Da die dezentrale Natur des WWW keinerlei Einschränkungen hinsichtlich der Formulierung von Vokabularen (RDF-Instanzen) und der zugehörigen Schemata (mit RDFS) trifft, ist das Entstehen heterogener „Beschreibungsinself“ abzusehen und inhärent unvermeidlich. Dies stellt ein prinzipielles Problem bei der Kommunikation dar, wie es bereits in [Hes72] beschrieben wurde.

Auch ein Lösungsansatz des sich abzeichnenden Dilemmas zwischen Freiheit der Formulierung einerseits und Verfügbarkeit für alle Kommunikationspartner andererseits wurde bereits in [Hes72] angedeutet. Der mehr formale Ansatz der Lösungsidee wird durch den W3C-Standard *Web Ontologien (OWL)* angestrebt.

Ein zentraler Schritt auf dem Weg zu maschinenverwertbarer Semantikbeschreibung ist die Verwendung von Vokabularen, wie sie durch RDFS formuliert werden können.

Werden im Vokabular zur Beschreibung der Grundrechenarten ausschließlich die Begriffe „Addition“, „Subtraktion“, „Multiplikation“ und „Division“ verwendet, so sind einerseits Synonyme wie „Zusammenzählen“, „Abziehen“ etc. von der maschinellen Verarbeitung ausgeschlossen. Andererseits können Vokabulare zusätzlich sicherstellen, dass alle Beschreibungen die gleiche Abstraktionsebene verwenden und ein Begriff auf einer höheren Abstraktionsebene – wie im Beispiel „ArithGrFkt“ (als Kurzform für „arithmetische Grundfunktion“) – nicht zusammen mit Begriffen einer weniger abstrahierten Beschreibungsebene („Addition“) verwendet wird.

Kombiniert man ein solches domänenspezifisches Vokabular von Begriffen mit einem Zusätzlichen von formalisierten Beziehungen zwischen den Begriffen, so entsteht eine *Ontologie*. Im Kern ist diese lediglich eine formale, maschinell auswertbare Definition der grundlegenden Konzepte eines Wissensgebietes sowie von Beziehungen zwischen diesen Konzepten [NM].

Damit können folgende Herausforderungen bei der Automatisierung von Diensten aus verschiedenen Domänen gelöst werden:

- ▼ Das Verknüpfen identischer Inhalte mit unterschiedlichen Identifikatoren: In unserem Beispiel kann durch die Verknüpfung (**equivalentClass**) aus Listing 4 geschlossen werden, dass jede **WSDLRef** auch eine **InterfaceRef** ist.
- ▼ Das logische Erschließen von neuen Zusammenhängen: Aufgrund der **isSubclassOf**-Deklarationen in Listing 4 kann durch logische Folgerung geschlossen (inferiert) werden, dass ein Dienst, der die **ArithGrFkt** beherrscht, auch multiplizieren kann.**
- ▼ Das Erschließen und Zusammenfassen von benötigten Grundfunktionalitäten, um einen geforderten Service zu leisten: Prinzipiell kann ein Dienstanutzer selbst Beziehungen zwischen den verwendeten Begriffen herstellen und so Schlussfolgerungen vorwegnehmen. So kann beispielsweise ein Nutzer, der die Potenzierungsoperation benötigt, auf die durch den Beispieldienst definierte Grundoperation **Multiplikation** zurückgreifen, wenn ihm die Information aus Listing 4 zur Verfügung steht.

Der Ausschnitt (die vollständige Fassung findet sich unter [jeckle.de]) des Listings 4 zeigt einen Ausschnitt einer Ontologie, die mit Hilfe des W3C-Standards *Web Ontology [WebOnt]* beschrieben wurde. Zusätzlich illustriert das Beschreibungsfragment das Ineinandergreifen mit der RDF-Beschreibung. Das Dokument liefert die URI-basierten Identifikatoren, mit

** Enthält eine Ontologie nur Beziehungen der Art **isSubclassOf**, dann spricht man i. d. R. von einer Taxonomie.



```

<owl:Class rdf:ID="&sws;Multiplikation">
  <rdfs:subClassOf rdf:resource="&sws;ArithGrFkt"/>
</owl:Class>
<owl:Class rdf:ID="InterfaceRef">
  <owl:equivalentClass rdf:resource="WSDLRef"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="WS-Artifact"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="WSDLRef">
  <owl:equivalentClass rdf:resource="&sws;WSDLRef"/>
  <rdfs:subClassOf rdf:resource="WS-Artifact"/>
</owl:Class>
<owl:Class rdf:ID="&ser;Potenzierung">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype="
        http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="&sws;isMultiplikation"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="&ser;höhereRechenarten"/>
  </rdfs:subClassOf>
</owl:Class>

<owl:ObjectProperty rdf:ID="&ser;isMultiplikation">
  <rdfs:domain rdf:resource="&ser;Potenzierung"/>
  <rdfs:range rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:ObjectProperty>

```

Listing 4: Ontologie-basierte Definition der Potenzierung als mehrfache Multiplikation

den ontologischen Beziehungsdefinitionen, welche anhand des Namensraumpräfixes `owl` kenntlich sind.

Das Beispiel versieht das RDF-Prädikat `isMultiplikation` des Subjekts `Potenzierung` mit der anwenderdefinierten Einschränkungsbziehung (`Restriction`), dass jede Potenzierung durch mindestens eine `Multiplikation` dargestellt werden kann. Der Sonderfall x_0 wurde hier aus Gründen der Übersichtlichkeit weggelassen.

Durch diesen Brückenschlag zwischen RDF-basierter Identifikation mittels URIs und der dezentralen Herstellung von Bedeutungszusammenhängen zwischen diesen können die zuvor isolierten Bedeutungsinselfen verknüpft werden.

Zusammenfassung

Im zweiten und dritten Artikel unserer Serie wurden die Konzepte aus der Welt des Semantic Web vorgestellt, die zukünftig eine entscheidende Rolle in Service-orientierten Architekturen spielen können. Bedeutung kann eindeutig mit Hilfe von RDF beschrieben werden. Die Einführung eines domänen-spezifischen kontrollierten Vokabulars erfolgt mittels RDF-Schema. Der Abgleich von Aussagen verschiedener Domänen kann über Ontologien (OWL) erfolgen.

Vokabularen und Ontologien. Diese müssen einschränkend genug sein, um ein eindeutiges Ergebnis liefern zu können. Andererseits sollten Vokabulare hinreichend flexibel sein, um erweiterbar und nicht unvereinbar mit anderen Vokabularen sind.

Noch zu klären bleibt, wie Semantik in eine Service-orientierte Architektur eingebettet werden kann. Diese Diskussion werden wir im nächsten und abschließenden Artikel wiederum am Beispiel der Web Services führen.

Literatur und Links

[DoJe04] W. Dostal, M. Jeckle, Semantik, Odem einer Service-orientierten Architektur, in: JavaSPEKTRUM, 1/04

[DoJeKr04] W. Dostal, M. Jeckle, Werner Kriechbaum, Semantik und Web Services: Beschreibung von Semantik, in: JavaSPEKTRUM, 2/04

[Hes72] H. Hesse, Das Glasperlenspiel, Suhrkamp, 1972

[jeckle.de] <http://www.jeckle.de/semanticWebServices/>

[NM] N. F. Noy, D. L. McGuinness, Ontology Development 101, http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

[RDFS] D. Brickley, R. V. Guha (Hrsg.), RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, 2004,

<http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>

[Sei80] L. Seiffert, Einführung in die Wissenschaftstheorie, C. H. Beck, München, 2002

[To103] Herr der Ringe, Teil 3 in der Verfilmung von P. Jackson

[WebOnt] D. L. McGuinness, F. van Harmelen (Hrsg.), OWL Web Ontology Language Overview, W3C Recommendation, 2004,

<http://www.w3.org/TR/2004/REC-owl-features-20040210/>

Jedoch ist es wichtig im Auge zu behalten, dass es sich bei der Einführung von Semantik nicht um die bloße Einführung einer neuen Technik handelt. Die eigentliche Herausforderung liegt in der „sinnvollen“ Definition von



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. Er ist Mitglied der Technical Advisory Group des W3Cs und vertritt die DaimlerChrysler-Forschung in verschiedenen Arbeitsgruppen. E-Mail: mario@jeckle.de.

Werner Kriechbaum arbeitet als IT-Architekt im Entwicklungslabor der IBM Deutschland. Schwerpunkte seiner Tätigkeit sind Performanz-Analyse und Optimierung von Anwendungen und die Beschreibung und Strukturierung komplexer Daten für Multimedia-Archive und biowissenschaftliche Anwendungen. E-Mail: kriechba@de.ibm.com.