

# Anfragesprachen für das Semantic Web

Joachim Baumeister und Dietmar Seipel

Aktuelles Schlagwort im **Informatik Spektrum**, 28 (1), 40-44, Springer, 2005

## Die Vision des Semantic Web

Das Semantic Web wird seit dem visionären Artikel von Berners-Lee, Hendler und Lassila [2001] allgemein als die nächste Evolutionsstufe des World Wide Web betrachtet. Denn, obwohl das WWW sicherlich als eine der größten technischen Erfindungen der letzten Jahrzehnte gefeiert werden kann, so ist es in seiner aktuellen Fassung leider nur manuell von Menschen zu bedienen. Die verfügbaren Informationen können daher oft nur unzureichend oder nicht genutzt werden, weil das Wissen meist über viele Web-Seiten verteilt ist und nicht maschinell integriert und interpretiert werden kann. Die semantische Annotation von Web-Inhalten und die hierbei gemeinsame Verwendung von Terminologien (Ontologien) gilt aus diesem Grund als die wesentliche Schlüsselerweiterung des Semantic Web.

Zwar entwächst das Semantic Web nur langsam seinen Kinderschuhen, trotzdem sind bereits grundlegende Technologien wie das Datenmodell oder Ontologie-Sprachen als Standard verfügbar. In einigen Nischen können heute schon erfolgreiche Anwendungen demonstriert werden, wie beispielsweise Wissensportale in firmenweiten Intranets oder spezialisierte elektronische Marktplätze. Die semantische Erweiterung von Suchmaschinen wird jedoch auch heute noch von vielen als die eigentliche Killer-Applikation des Semantic Web betrachtet.

All diese Anwendungen erfordern einen effizienten Anfragemechanismus an strukturierte Massendaten. Hieraus motiviert sich die Entwicklung von deklarativen Sprachen, welche Anfragen an die Inhalte, aber auch an deren semantische Annotation ermöglichen. In Anlehnung an die Erfolge in der Datenbankforschung ist es das Ziel, eine portable Hochsprache (in der Tradition von SQL) bereitzustellen, mit der Anfragen schnell und einfach formuliert werden können.

Dieser Beitrag wird kurz die beiden aktuellen Standards zur Beschreibung und Annotation von Semantic Web-Inhalten – RDF(S) und OWL – vorstellen und die Anforderungen an semantische

Anfragesprachen beschreiben. Zum aktuellen Zeitpunkt ist noch keine standardisierte Anfragesprache verfügbar. Die vorgestellte RDF(S)-Anfragesprache RQL beinhaltet aber alle wesentlichen Züge einer Semantic Web-Anfragesprache. Außerdem wird kurz auf Anfragen und Analysen von OWL-Dokumenten eingegangen.

## Semantic Web-Sprachen

Aufbauend auf der Sprache HTML, welche momentan der bestehende Standard für die Beschreibung von Web-Inhalten ist, treten immer häufiger XML-basierte Sprachen, wie beispielsweise XHTML, auf. So gilt XML heute als universelle Meta-Sprache, welche gestützt auf eine Vielzahl von Werkzeugen (Parser, Editoren, etc.) für den Datenaustausch zwischen Applikationen genutzt wird.

Problematisch ist jedoch, dass XML per se keine Unterstützung für die Beschreibung der Semantik der formulierten Daten bietet. Ein einfaches Beispiel ist die fehlende Bedeutung des Zusammenhangs zwischen Tag-Elementen, welche ineinander geschachtelt sind: Meist ist diese nur implizit in der jeweiligen Anwendung verankert.

Das Resource Description Framework (RDF) ist eine domänen-unabhängige Sprache zur semantischen Beschreibung von Datenmodellen [RDF]; RDF wird üblicherweise in XML repräsentiert. Das jeweils anwendungsspezifische Vokabular des RDF-Datenmodells wird im RDF Schema (RDFS) definiert. Die Analogie zu XML Schema trifft bei RDF Schema nicht zu, da XML Schema die Struktur von XML-Dokumenten durch Constraints beschreibt, während im RDF Schema das Vokabular des Datenmodells generiert wird, d.h. es werden Eigenschaften zwischen Objekten und Wertebereichen sowie Beziehungen zwischen einzelnen Objekten definiert.

Im Rahmen dieses Artikels werden wir als Beispiel einen kleinen Auszug aus einer Drucker-Anwendung von Hewlett Packard verwenden, welche in Abbildung 1 in UML-Notation dargestellt ist.

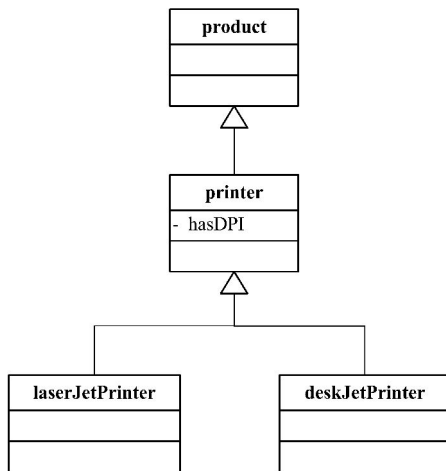


Abbildung 1: Ein Ausschnitt der Drucker-Ontology in UML

Die textuelle Repräsentation in RDF(S) könnte in einer XML-basierten Darstellung wie folgt aussehen:

```

<rdfs:Class rdf:ID="product"/>
<rdfs:Class rdf:ID="printer">
  <rdfs:subClassOf
    rdf:resource="#product"/>
</rdfs:Class>
<rdfs:Class rdf:ID="laserJetPrinter">
  <rdfs:subClassOf
    rdf:resource="#printer"/>
</rdfs:Class>
<rdfs:Class rdf:ID="deskJetPrinter">
  <rdfs:subClassOf
    rdf:resource="#printer"/>
</rdfs:Class>
<rdf:Property rdf:ID="hasDPI">
  <rdfs:domain
    rdf:resource="#printer"/>
  <rdfs:range
    rdf:resource="#&rdf;Literal"/>
</rdf:Property>
  
```

Vereinfacht betrachtet ist die Ausdrucksmächtigkeit von RDF(S) beschränkt auf die Bildung von Hierarchien über Klassen und Eigenschaften sowie auf die Definition von Wertebereichen und Domainen. So konnte die Web Ontology Working Group des W3C schon bald einige Anwendungsfälle identifizieren, für welche die Ausdrucksmächtigkeit von RDF(S) nicht ausreichte [WebOnt].

Die Forschungsergebnisse der Modellierungssprachen DAML und OIL für Ontologien wurden schließlich als Ausgangspunkt für die Ontology Web Language (OWL)

herangezogen. OWL baut auf RDF(S) auf und erweitert die Ausdrucksmöglichkeit für die semantischen Begrifflichkeiten in vielen Punkten, beispielsweise durch die Möglichkeit disjunkte Konzepte zu definieren oder durch die Fähigkeit neue Klassen aus der booleschen Kombination (Vereinigung, Schnitt, Komplement) bereits bestehender Klassen zu erstellen.

Da eine erhöhte Ausdrucksmächtigkeit immer mit einer reduzierten Unterstützung für eine effiziente Schlussfolgerungsfähigkeit einhergeht, entschloss man sich, die Sprache OWL in drei Ausbaustufen zu skalieren: OWL Lite, OWL DL, und OWL Full. Eine Einführung in Semantic Web Sprachen kann man bei Antoniou und van Harmelen [2004] finden.

## Herausforderungen für Anfragen an Semantic Web-Inhalte

Semantic Web-Sprachen werden überwiegend in XML repräsentiert, und daher wäre es ein nahe liegender Gedanke bereits standardisierte Anfragesprachen für XML-Dokumente wie beispielsweise XQuery zu verwenden; eine Einführung in XQuery ist bei Lehner und Schöning [2004] zu finden. Durch die Universalität von XML können solche Anfragesprachen nur Anweisungen formulieren, welche sich an der Syntax der jeweiligen XML-Dokumente orientieren. Eine geeignete Anfragesprache für Semantic Web-Dokumente muss aber in der Lage sein, das Datenmodell von RDF/OWL und die semantischen Eigenschaften des Vokabulars zu verstehen. Zum Beispiel sollte eine Anfrage über alle Instanzen von `printer` nicht nur Instanzen von `printer` berücksichtigen, sondern auch alle Unterklassen des Konzeptes `printer`, wie `laserJetPrinter` oder `deskJetPrinter`. Wie in diesem einfachen Beispiel zu erkennen ist, benötigen Semantic Web-Anfragesprachen also die Fähigkeit zu Deduktion.

Eine weitere, sehr viel schwierigere Anforderung stellt die Möglichkeit zur Integration dar. Semantic Web-Inhalte werden in einer realen Umgebung nicht auf einer gemeinsamen Ontologie basieren. Vielmehr

wird es spezialisierte Ontologien für die Inhaltsbeschreibung in den jeweiligen Anwendungsbereichen geben, die sich durchaus in allgemeinen Begrifflichkeiten überschneiden können. Typischerweise werden Anfragen an das Semantic Web nicht auf ein Dokument begrenzt sein, sondern Antworten liefern, die aus der Aggregation von mehreren Dokumenten bestehen. Insofern ist die semantische Integration von Dokumenten bzw. von den zugrunde liegenden Ontologien von wesentlicher Bedeutung. Stuckenschmidt [2003] diskutiert neben den Aspekten der Deduktion und der Integration zusätzliche Herausforderungen für Anfragesprachen, wie zum Beispiel die Möglichkeit der Approximation, der Transformation und der Entdeckung.

## Anfragen an RDF(S)-Dokumente mit RQL

Zum aktuellen Zeitpunkt existiert eine Reihe von konkurrierenden Vorschlägen für eine Anfragesprache für das Semantic Web; darunter fallen beispielsweise RQL, SPARQL, SeRQL, RdfDB, Algae, Edutella, RDFQL, SquishQL, Inkling, RDQL, RDFStore, Triple, N3, Versa, und DQL. Im Folgenden werden wir RQL [Karvounarakis et al. 2002] mit Hilfe einiger typischer Anfragen skizzieren, da RQL die grundlegenden Konzepte für Semantic Web-Anfragesprachen implementiert und bereits in einigen Anwendungen verwendet wird. Beispielsweise sind RSSDB und Sesame, zwei Semantic Web-Managementsysteme, die Teile von RQL unterstützen. RQL ist eine typisierte, funktionale Sprache, welche generalisierte Pfadausdrücke sowohl über Knoten als auch über Kanten versteht. Mit RQL können sowohl RDF- als auch RDFS-Daten einheitlich angefragt werden.

### Grundlegende Operationen

Für die Traversierung von Klassenhierarchien bietet RQL die Funktionen `subClass^(X)` und `subClass(X)` zur Bestimmung der direkten Subklassen bzw. transitiven Subklassen der Klasse `X`; analog existieren die Funktionen `subProperty^(Y)` und `subProperty(Y)`. Mit der grundlegenden Anfrage `laserJetPrinter` würde eine Sammlung mit allen Instanzen der entsprechenden Klasse zurückgeliefert werden. Über Sammlungen des gleichen Types können mit `union`, `intersect` und `minus` einfache Mengenoperationen ausgeführt werden.

Ebenso erzeugen die Befehle `seq` und `bag` über beliebigen Anfrageergebnissen wieder neue Sequenzen und Sammlungen; zum Beispiel liefert die Anfrage

```
seq(domain(hasDPI), range(hasDPI))
```

eine Sequenz mit der Domäne und dem Wertebereich der Eigenschaft `hasDPI`.

Zur Filterung der Anfrageergebnisse bietet RQL einfache Boolesche Prädikate wie `=`, `<`, `>` und `like`. Während `like` zum Vergleich von Zeichenketten gedacht ist, können die übrigen Prädikate auf beliebigen Literalen angewandt werden; interessanterweise auch zum Vergleich von Klassenbeziehungen: So liefert

```
printer > laserJetPrinter
```

den Wert `true`, da `laserJetPrinter` als eine Unterklasse von `printer` modelliert ist. Weiterhin werden dem Sprachschatz von RQL eine Anzahl von Funktionen zur Aggregation von Ergebnissen beigelegt (`min`, `max`, `avg`, `sum`, `count`).

Ähnlich zu SQL wird aufbauend auf den obigen Basisanfragen ein `select-from-where`-Konstrukt angeboten, welches es ermöglicht auf den Ergebnissen der Basisanfragen (d.h. auf Sammlungen von RDF(S)-Daten) filternd zu operieren.

### Anfragen an RDF Schema

Durch die äußerst freie und anwendungsspezifische Erstellung von RDF Datenmodellen ist es häufig sinnvoll, auch Anfragen an das zugrundeliegende RDF Schema zu stellen. So würde folgende Anfrage alle Klassen `%C1` bzw. Werte `%C2` auflisten, welche die Eigenschaft `hasDPI` als Domäne bzw. als Wertebereich haben:

```
select %C1, %C2
from {%C1}hasDPI{%C2}
```

Im `from`-Teil der Anfrage erkennt man die Notation `{}` zur Definition von Variablen über Klassen, welche standardisiert mit einem `%`-Zeichen beginnen. Die abkürzende Notation `{%C1}hasDPI{%C2}` bedeutet eine Iteration über der Domäne bzw. dem Wertebereich von `hasDPI`, ist also äquivalent zu dem Ausdruck

```
%C1 <= domain(hasDPI) and
%C2 <= range(hasDPI).
```

Wie bei SQL dient der `select`-Teil zur Projektion der gewünschten Ergebnisse. Gewöhnlich wird eine Sequenz dieser Ergebnisse erzeugt, und mit dem `*`-Operator werden alle Variablen aus der `from`-Klausel in die Sequenz aufgenommen.

Eine weitere typische RDFS-Anfrage könnte wie folgt lauten: Zeige alle Eigenschaften der Klasse `printer` mit den zugehörigen Wertebereichen an:

```
select @P, range(@P)
from {$C}@P
where $C = printer
```

In der `from`-Klausel der Anfrage erkennt man die `@`-Notation, zur Einführung von Variablen über Eigenschaften von Klassen: So wird hier in der Variable `@P` über allen Eigenschaften der Klasse `$C` iteriert; im `where`-Teil wird diese Variable auf die Klasse `printer` eingeschränkt.

## Anfragen an RDF-Daten

In RQL können generalisierte Pfadausdrücke frei aus Knoten- und Kantenbeschreibungen zusammengesetzt werden, welche sowohl aus RDF-Daten als auch aus RDFS stammen können.

Das zu SQL analoge Konzept des Joins wird durch die Punkt-Notation eingeführt. Folgende Anfrage liefert alle Drucker mit mehr als 600 DPI:

```
select P, D
from printer{P}.hasDPI{D}
where D > 600
```

In der `from`-Klausel wird die Datenvariable `P` an alle Instanzen der Klasse `printer` gebunden, der Wertebereich von `hasDPI` wird an `D` gebunden, und deren Instanzen geeignet in einem Join konkateniert. Der Ausdruck ist damit äquivalent zu

```
printer{P1}, {P2}hasDPI{D}
where P1 = P2
```

In der `where`-Klausel wird die gebundene Variable `D` entsprechend weiter eingeschränkt. Darüber hinaus bietet RQL die Möglichkeit gemischte Anfragen zu stellen, welche sowohl das RDF-Datenmodell als auch das RDF Schema berücksichtigen.

## Anfragen an OWL-Dokumente

OWL umfasst RDF(S) und es bietet darüber hinaus noch viele weitere Möglichkeiten zur Repräsentation von Metadaten.

Das folgende OWL-Element drückt z.B. aus, dass die Klasse `hpProduct` aus allen Produkten besteht, welche ausschließlich von `HewlettPackard`

angefertigt werden.

```
<owl:Class rdf:ID="hpProduct">
  <rdfs:comment>
    HP products are exactly
    those products that are
    manufactured by Hewlett Packard
  </rdfs:comment>
  <owl:intersectionOf>
    <owl:Class rdf:about="#product"/>
    <owl:Restriction>
      <owl:onProperty>
        rdf:resource=
          "#manufactured_by"/>
      <owl:hasValue>
        <xsd:string
          rdf:value=
            "HewlettPackard"/>
        </owl:hasValue>
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
```

Dies wird dadurch erreicht, dass der Wert der Eigenschaft `manufactured_by` mittels `hasValue` auf `HewlettPackard` eingeschränkt wird. Auch komplexere Bedingungen, welche verlangen, dass alle Werte einer Eigenschaft oder mindestens ein solcher Wert in einer anderen Klasse liegen bzw. liegt, können mit Hilfe von mächtigen `allValuesFrom`- bzw. `someValuesFrom`-Konstrukten leicht ausgedrückt werden. Eine äquivalente Repräsentation von OWL-Dokumenten kann in Beschreibungslogiken oder in einer LISP-artigen Syntax erfolgen.

In OWL-Systemen wie RACER kann man (Lisp-artige)Anfragen an solche Dokumente stellen. Diese Anfragen können z.B. alle Instanzen oder Unterklassen einer vorgegebenen Klasse wie `hpProduct` ermitteln.

Seipel und Baumeister [2004] beschreiben Anfragen und Analysen für OWL-Wissensbasen (wie zum Beispiel an eine Erweiterung der obigen `printer`-Ontologie) mit Hilfe der deklarativen XML-Anfragesprache `FNQUERY`. Diese Sprache ist logikbasiert und sie erlaubt beliebige Deduktionen durchzuführen. Zur Evaluation von OWL-Wissensbasen wurde `FNQUERY` um Regeln erweitert, so dass Schlussfolgerungen über dem OWL-Modell gezogen werden können.

Ähnlich dazu wurden die Anfrage- und Transformationssprachen `Xcerpt` und `visXcerpt` [Berger et al. 2004] im Hinblick auf Standard Web- und Semantic Web-Anwendungen

entwickelt. Sie sind Zwillingssprachen (die eine textuell, die andere visuell), welche beide auf denselben Paradigmen und Prinzipien basieren.

## Ausblick

Das Semantic Web als nächste Evolutionsstufe des WWW ist eine interessante Herausforderung für die gesamte Informatik. Nachdem die Modellierungssprachen von Semantic Web-Inhalten durch RDF(S) bzw. OWL weitgehend standardisiert sind, treten neben der Integration von und dem Schlussfolgern über solchen Inhalten auch die Frage nach geeigneten Anfragesprachen für Semantic Web-Inhalte auf. Bedauerlicherweise sind hier die Standardisierungsbemühungen noch nicht sehr weit vorangeschritten und eine rege Forschung bringt kontinuierlich neue Anregungen. Neben der Anfragesprache muss auch die damit zusammenhängende Infrastruktur für RDF/OWL-Daten diskutiert werden. Momentan entsteht eine Vielzahl von Managementsystemen für Semantic Web-Inhalte. Das rege Open Source-Projekt Sesame ist nur ein Beispiel; es macht sehr schön deutlich, wie sich in Zukunft diese Forschungsrichtung weiter entwickeln wird: Auf der einen Seite ist die Optimierung von Ausdrücken zu einer effizienteren Abarbeitung von Anfragen ein Gegenstand der Forschungsbemühungen. Auf der anderen Seite werden aber auch Konzepte untersucht, welche Anfragen an die im Vergleich zu RDF(S) mächtigere Sprache OWL ermöglichen. Einige Schritte in diese Richtung sieht man beispielsweise in der Anfragesprache SeRQL des Systems Sesame.

## Literatur

- Antoniou, G., Van Harmelen, F.: A Semantic Web Primer (Cooperative Information Systems), MIT Press 2004
- Berger, S., Bry, F., Bolzer, O., Furche, T., Olteanu, D., Schaffert, S., Wieser, C.: *Xcerpt and visXcerpt: Twin Query Languages for the Semantic Web*. 3rd International Semantic Web Conference, Demonstration, 2004
- Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web, Scientific American 2001
- Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema, Proceedings of the 1st International Semantic Web Conference, Springer LNCS 2342, 54-68, 2002
- Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M.: RQL: A Declarative Query Language for RDF, Proc. 11th Intl. World Wide Web Conference (WWW'02), Honolulu, Hawaii, USA, May 7-11, 2002
- Lehner, W., Schöning, H.: XQuery - Grundlagen und fortgeschrittene Methoden, dpunkt.verlag 2004
- RDF: <http://www.w3.org/RDF/>
- Seipel, D., Baumeister, J.: Declarative Methods for the Evaluation of Ontologies, Künstliche Intelligenz 4/2004, 51-57, 2004
- Sesame: <http://www.openrdf.org/>
- Stuckenschmidt, H.: Query Processing on the Semantic Web, Künstliche Intelligenz 3/2003, 22-26, 2003
- WebOnt: <http://www.w3c.org/2001/sw/WebOnt>

### Kontakt:

Joachim Baumeister, Dietmar Seipel  
Institut für Informatik  
Am Hubland  
97074 Würzburg  
email: [baumeister/seipel@informatik.uni-wuerzburg.de](mailto:baumeister/seipel@informatik.uni-wuerzburg.de)