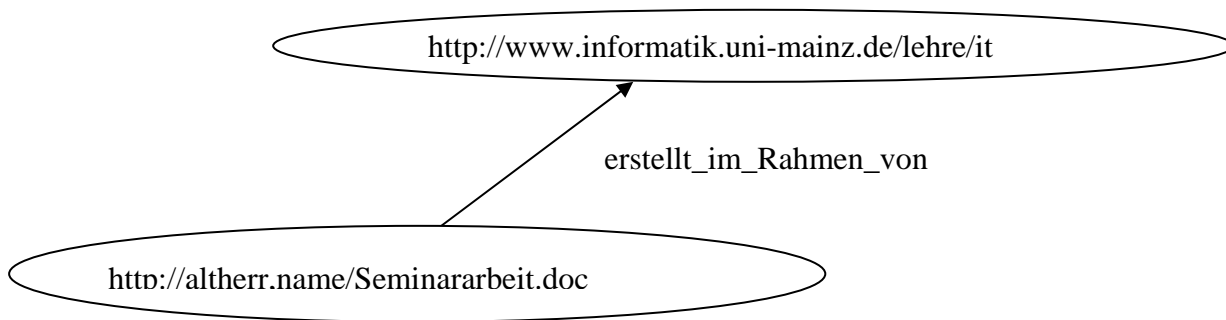


# Semantic Web

---



Seminar Internet-Technologie

Seminarleitung:  
Prof. Dr. Herbert Göttler,  
Thomas Gottron

Seminararbeit von Florian Altherr WS 2005/06

## **Inhaltsverzeichnis**

<b>1. Einführung</b>	<b>Seite 3</b>
1.1 Begriffsklärung	
1.2 Die Idee des Semantic Web - Motivation und Vision	
1.3 Voraussetzungen für die Umsetzung	
1.4 Die Grenzen von XML	
<b>2. Das Schichtenmodell</b>	<b>Seite 8</b>
2.1 Basistechnologien: Unicode, URI	
2.2 XML	
2.3 RDF und RDFS	
2.4 Ontologien, OWL und seine Vorgänger	
2.4.1 Syntaxbeispiele	
2.4.2 Umfassendes Beispiel: Bürgeramt.owl	
2.5 Logic, Proof, Trust	
<b>3. Anwendungsbeispiele</b>	<b>Seite 15</b>
3.1 Semantic Web Services	
<b>4. Fazit und Ausblick</b>	<b>Seite 17</b>
<b>5. Quellenverzeichnis</b>	<b>Seite 18</b>

## 1. Einführung

Der Begriff des »Semantic Web« existiert seit circa 1998 und wurde maßgeblich von Tim Berners-Lee<sup>1</sup>, dem Direktor des W3C und Erfinder des WWW, geprägt. Nachdem die Thematik schon 1996 in der Diskussion um die HTML-basierte Ontologie-Beschreibungssprache SHOE aufgekommen war, äußerte sich Berners-Lee 1998 in seiner "Semantic Web Roadmap"<sup>2</sup> grundsätzlich dazu. Er sieht die Zukunft des Webs in einer Erweiterung und logischen Weiterentwicklung, die er als Semantic Web bezeichnet und die effizientere Online-Recherchen und die Automatisierung vieler Informations- und Kommunikationsprozesse ermöglichen soll. Sogenannte intelligente Software-Agenten sollen dem Menschen bestimmte Aufgaben abnehmen und ihm helfen, sich im Informationsdschungel des WWW zurechtzufinden.

Die vorliegende Arbeit beschreibt Idee und Motivation sowie zentrale Konzepte des Semantic Web und zeigt auf, welche Gedanken- und Entwicklungsschritte zu seiner Realisierung notwendig sind. Zum Schluss wird aufgezeigt wie weit das Semantic Web bisher konkret gediehen ist und welche Anwendungsbeispiele mit Semantic Web-Technologien umgesetzt werden können.

### 1.1 Begriffsklärung

Unter **Semantik** versteht man ein Teilgebiet der Linguistik, das sich mit Sinn und Bedeutung von Sprache bzw. sprachlichen Zeichen beschäftigt. Im Semantic Web sollen also Maschinen die Bedeutung von Daten verstehen. Dies erfolgt durch Anreicherung der im Web vorhandenen Daten mit Metainformationen. Es handelt sich dabei um die zentrale Idee, die dem Semantic Web zugrunde liegt und die man mit den im Folgenden beschriebenen Technologien umzusetzen sucht.

Theoretisch-informatischer Hintergrund ist die Möglichkeit der Repräsentation von Wissen durch **semantische Netze**. Begriffe und Relationen werden hierbei durch einen verallgemeinerten Graphen modelliert. Dabei stellen die Knoten des Graphen die Begriffe dar. Beziehungen zwischen den Begriffen werden durch die Kanten des Graphen abgebildet. Solche semantischen Netze mit einer eingeschränkten Menge von Relationen sind zum Beispiel Taxonomien oder Thesauri, die vor allem Synonyme, aber auch Ober- und Unterbegriffe verwalten. Die wörtliche Übersetzung von Semantic Web weist zwar auf dessen Ursprung in der Theorie der semantischen Netze hin, allerdings handelt es sich hierbei um eine konkrete Spezifikation eines lose verteilten semantischen Netzes.

Beim **Semantic Web** handelt es sich also um eine Erweiterung des WWW um die Dimension der Bedeutung, um ein Web der 2. Generation sozusagen: Die Vision von Tim Berners-Lee für die Entwicklung des WWW sind Daten, die für Maschinen verständlich sind.

---

<sup>1</sup> <http://www.w3.org/People/Berners-Lee/>.

<sup>2</sup> <http://www.w3.org/DesignIssues/Semantic.html>

## 1.2 Die Idee des Semantic Web - Motivation und Vision

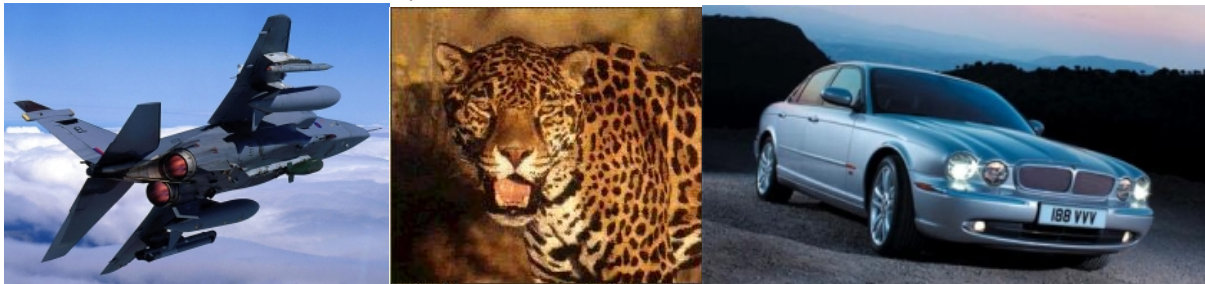
Das semantische Web ist nicht als Revolution, sondern als evolutionäre, logische Weiterentwicklung des World Wide Web aus und mit den bereits existierenden Techniken anzusehen. Ausgangspunkt der Überlegungen ist, die riesige Menge an im Web vorhandener Information besser nutzbar zu machen. Maschinen sollen dem Menschen dabei helfen, aus der Informationsflut relevantes Wissen zu extrahieren und zu verwerten. Dies soll im Folgenden näher ausgeführt werden.

In den letzten 2 Jahrzehnten ist das WWW zu einem unverzichtbaren Medium für Millionen von Menschen geworden, die es beruflich oder privat nutzen, um sich Wissen anzueignen, Informationen zu suchen, Daten und Dokumente auszutauschen oder miteinander zu kommunizieren. Dieser Informationsfluss funktioniert jedoch nur deshalb, weil am Ende der Kommunikationskette der Mensch steht, der die Daten interpretiert und sie einzuordnen weiß.

Das heutige Internet ist vor allem für den menschlichen Nutzer konzipiert und damit repräsentationsorientiert. Ziel ist, Informationen in einfachem und übersichtlichem Layout visuell möglichst ansprechend darzustellen, so dass sie leicht zugänglich und intuitiv erfassbar sind. Aus dieser Repräsentationsorientierung für menschliche Nutzer ergeben sich aber Probleme für die intermaschinelle Kommunikation: Ein menschlicher Nutzer kann aus den Texten, Bildern oder generell Inhalten einer Webseite ihre Bedeutung meist sehr einfach erfassen, eine Maschine aber nicht. Menschen denken assoziativ, d.h. in Zusammenhängen, sie erkennen, ob eine Information in einen bestimmten Kontext gehört oder nicht, relevant oder irrelevant ist.

Allerdings ist gerade die Informationssuche eine zeitintensive, oft uneffektive, intellektuell wenig anspruchsvolle Angelegenheit, die man gerne an Maschinen delegieren würde. Jemand, der beispielsweise den günstigsten Preis für ein Auto herausfinden will, muss die Web-Seiten sämtlicher Anbieter durchforsten, jeweils den Preis heraussuchen und schließlich die gefundenen Preise vergleichen, also etliche Male die gleichen, stupiden Handlungsschritte durchführen. Ein Software-Agent könnte ihm diese und viele weitere Aufgaben abnehmen, jedoch liegt genau hier das Problem: Maschinen bleiben die auf den Seiten des WWW vermittelten Informationen weitgehend verschlossen. Zwar können sie Web-Inhalte lesen und nach Schlüsselwörtern durchsuchen, es mangelt aber an einem wirklichen Verständnis des Gelesenen in unserem Sinne. Daher können sie keine Zusammenhänge herstellen, inhaltliche Ähnlichkeiten finden usw.

Befände sich ein Autokäufer z.B. auf der Suche nach einem Jaguar, wäre der erste Schritt des Agenten, Anbieterseiten ausfindig zu machen. Schon an dieser Stelle hätte er das Problem, relevante Seiten herauszufiltern.



3

---

3

<http://images.google.com/images?client=opera&rls=en&q=Jaguar&sourceid=opera&ie=utf-8&oe=utf-8&sa=N&tab=wi>

So verdeutlichen obige Bilder aus den zurück gelieferten Seiten (hier anhand einer Suche mit Google), dass der Softwareagent nicht erkennen würde, dass es sich um Homonyme handelt bzw. welches der Homonyme das für den Benutzer interessante ist. Ebenso wenig wäre er in der Lage, Synonyme bei der Suche mit einzubeziehen (Handy/ Mobiltelefon).

Vorausgesetzt, der Agent wäre fündig geworden, so wäre spätestens der nächste Schritt, auf der Anbieterseite den Preis heraus zu finden, eine schier unlösbare Aufgabe, da jeder Anbieter ein eigenes Layout verwendet. Der Agent wüsste zudem nicht, ob er eine Suchfunktion benutzen kann, wenn ja, wie und mit welchen Parametern. Dazu kämen viele weitere technische Hindernisse.

Diese Probleme des Nichterkennens der inhaltlichen Bedeutung von Webseiten setzen also der heutigen Nutzung des WWW für Maschinen enge Grenzen, ihre Behebung ist Forschungsgegenstand von Wissenschaftlern weltweit. Abhilfe soll hier das Semantic Web schaffen, dessen Erfinder und Initiatoren sich zum Ziel gesetzt haben, das bestehende Web um semantische Informationen zu erweitern und damit softwaretechnisch nutzbar zu machen.

Oder wie Tim Berners-Lee seine Vision formuliert: heute existiert ein Web mit Daten, die für Maschinen lesbar sind, die Zukunft wird ein Web mit Daten sein, die für Maschinen verständlich sind. Informationen sollen also zusätzlich zu der für Menschen lesbaren Form auch formal, in einer für Maschinen verarbeitbaren Form repräsentiert werden, damit Programme darauf operieren können.

In einem viel beachteten Artikel im Scientific American formulierten Tim Berners-Lee, James Hendler und Ora Lassila schließlich 2001 die These für ein breiteres Publikum und definierten ihre Vision folgendermaßen:

„The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation“<sup>4</sup>

D.h. die Grundidee des Semantic Web ist eine Erweiterung der gegenwärtigen Form des Web, nach der Informationen mit einer wohldefinierten Bedeutung versehen werden, um die verbesserte Zusammenarbeit zwischen Mensch und Maschine zu ermöglichen.

So ließen sich dann komplexe Aufgaben wie die folgende bewältigen:

„Buche eine 7-tägige Reise nach Nizza, suche nach einem Hotel, einem Flug und einem Mietauto, reisen kann ich zwischen dem 25.02. und dem 24.4. 2006, ich will möglichst wenig zahlen, aber per Auto oder Bus höchstens 10 Min. brauchen zum Strand“

Der Agent würde dabei persönliche Vorlieben und anderes lokales Wissen mit im Web vorhandenem Wissen verknüpfen. Zusätzlich würde er Wissen aus der Vergangenheit mit einbeziehen.

---

<sup>4</sup> Tim Berners-Lee, James Hendler und Ora Lassila: The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, Scientific American, May 17, 2001

### 1.3 Voraussetzungen für die Umsetzung

Damit ein Agent wie obiger im Semantic Web erfolgreich agieren, d.h. Daten verstehen und interpretieren kann, gibt es generell zwei mögliche Wege. Entweder es müssen alle Daten mit semantischer Zusatzinformation angereichert werden, so genannten Metadaten, die den Agenten die Interpretation abnehmen. Oder aber man müsste Maschinen mit genügend künstlicher Intelligenz ausstatten, die ihnen die Fähigkeit verleihe, die Bedeutung von Informationen in unserem Sinne zu erkennen.

Nach Vorstellung von Tim Berners-Lee und dem WWW-Konsortium, das die Entwicklung an Semantic Web-Technologien koordiniert und standardisiert, ist die erste Variante der richtige Weg: Daten werden möglichst ausführlich mit Metadaten beschrieben, wobei diese so genau spezifiziert und strukturiert sein müssen, dass Agenten sie nicht nur korrekt interpretieren, sondern auch verstehen können. „Verständnis“ heißt hier, dass Maschinen nicht nur in die Lage versetzt werden, die Bedeutung von (Meta-)Daten zu erfassen, sondern auch, aus vorhandenen Daten Schlüsse zu ziehen und auf diese Weise neue Informationen zu erzeugen“.<sup>5</sup>

Die zentrale Rolle spielt vor allem die Modellierung von Beziehungen zwischen Daten-Ressourcen, die die Bedeutung einzelner Ressourcen wesentlich deutlicher macht als die Beschreibung ihrer Eigenschaften.

Zusammenfassend lässt sich sagen, dass es sich beim Semantic Web um eine metadatenbasierte Infrastruktur für Wissensrepräsentation und Schlussfolgerungen im Web handelt mit wohldefinierten Operationen auf wohldefinierten Daten und nicht um Künstliche Intelligenz, wengleich Techniken aus dem Forschungsbereich der KI zum Einsatz kommen (Logik, Ontologien, WBS). Dies macht auch die Notwendigkeit von Standards deutlich, die im Folgenden erläutert werden.

### 1.4 Die Grenzen von XML

Bei der Suche nach einem geeigneten Datenmodell stellt sich zunächst die Frage warum man nicht XML verwenden kann, welches derzeit einen wahren Hype erfährt. Es gilt als selbstbeschreibend und könnte somit möglicherweise das Problem der fehlenden Semantik in Dokumenten lösen. Für den Aufbau von Vokabularen mit XML gibt es sogar eine vom W3C standardisierte Sprache – XML-Schema.

In vielen Branchen findet man Versuche einer Normierung dieser Art. Branchen legen sich mithilfe eines XML-Schemas auf ein Austauschformat fest. Dabei ist allerdings oft ein „Wildwuchs“<sup>6</sup> zu beobachten, der die Kommunikation stört. Diese Probleme, die durch Unterschiede in der Terminologie entstehen, soll ein Beispiel verdeutlichen:

Nachfolgende Abbildung zeigt das Angebot eines Weinhändlers, welches er über das Netz verbreitet. Es beschreibt seinen Weinladen, bestehend aus Laden und Lager, indem er 25 Kisten eines bestimmten Chiantis zu einem bestimmten Preis anbietet. Will nun ein Kunde „eine Kiste Rotwein des Jahrgangs 1995“ über seinen Agenten besorgen lassen, so treten mehrere Probleme auf:

---

<sup>5</sup> Notholt, Jochen: Die Standards des Semantic Web (Teil 2) 2005  
<http://www.jurpc.de/aufsatz/20050065.htm>, Stand 10.02.2006

<sup>6</sup> Mintert, Stefan: Das semantische Web. Abgehoben. In: iX-Special 1/2004, Seite 124

Der Agent sichtet das Angebot, findet aber weder „Jahrgang“ noch „Rotwein“. Er erkennt auch nicht, dass Jahr und Jahrgang Synonyme sind und Chianti ein Rotwein ist.

```

- <Weinladen>
  - <Laden>
    <Tel href="tel:+4961314711">06131/4711</Tel>
    <Oeffnungszeiten>Mo-Fr 19-16 Uhr</Oeffnungszeiten>
  </Laden>
  - <Lager>
    - <Italien>
      - <Chianti>
        <Weingut>...</Weingut>
        <Jahr>1995</Jahr>
        <PreisProFlasche>20</PreisProFlasche>
        <Lagermenge>25 Kisten</Lagermenge>
      </Chianti>
    </Italien>
    <Frankreich> </Frankreich>
  </Lager>
</Weinladen>

```

Vorwegnehmend gesagt ließen sich im W3C-Szenario solche Probleme mithilfe einer so genannten Ontologie lösen, in diesem Fall einer, die Weine klassifiziert. Sie vermittelt dem Agenten das Wissen darüber, dass die Klassen Wein, Rotwein und Chianti existieren und in welcher Beziehung sie zueinander stehen. So weiß er, dass Rotwein eine Unterklasse von Wein und Chianti eine Unterklasse von Rotwein darstellt. Die Ontologie, hier in der Web Ontology Language (OWL) ausgedrückt, ermöglicht ihm, das Angebot in die nähere Auswahl zu nehmen.

Der Agent weiß:

```

<owl:Class rdf:ID="Wein"/>

[ ] <owl:Class rdf:ID="Rotwein">
  <rdfs:subClassOf rdf:resource="#Wein"/>
</owl:Class>

[ ] <owl:Class rdf:ID="Chianti">
  <rdfs:subClassOf rdf:resource="#Rotwein"/>
</owl:Class>

```

Auch das zweite Problem, der unklaren Begrifflichkeit von „Jahrgang“ und „Jahr“ ließe sich mit Semantic Web-Technik problemlos lösen, indem in OWL die Datentypereigenschaften „Jahrgang“ und „Jahr“ als äquivalent gekennzeichnet werden.

Der Agent weiß:

```

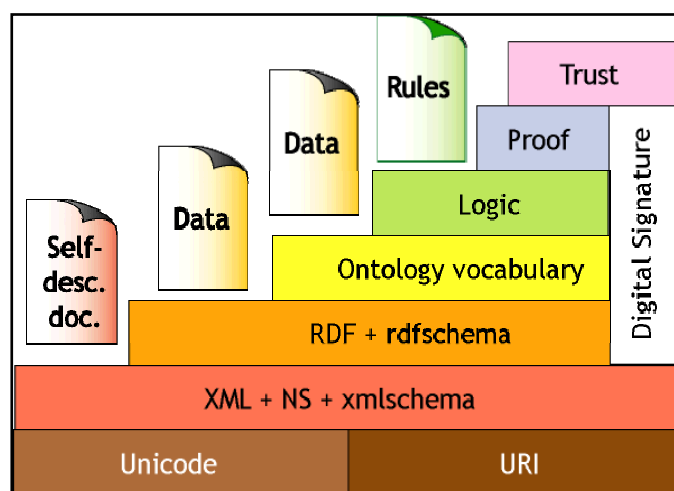
[ ] <owl:DatatypeProperty rdf:ID="Jahrgang">
  <owl:equivalentProperty rdf:resource="#Jahr"/>
  <rdfs:domain rdf:resource="#Wein"/>
  <rdfs:range rdf:resource="&xsd;#integer"/>
</owl:DatatypeProperty>

```

Das Beispiel des Weinagenten zeigt die Grenzen von XML auf und ebenso die Möglichkeiten, die Semantic Web-Techniken bieten. Statt „der Welt Standard-XML-Vokabulare für jeden Zweck überzustülpen“<sup>7</sup>, ist es möglich, dezentrale Vokabulare zu schaffen, die wiederverwendbar sind und bereits bestehende erweitern können.

## 2. Das Schichtenmodell

Diese Standards zu formulieren und deren Entwicklung voranzutreiben, hat sich das W3C, speziell die Gruppe der Semantic Web Activity<sup>8</sup>, zur Aufgabe gemacht. Die festgelegten Techniken lassen sich anschaulich darstellen im von Berners-Lee entwickelten Schichtenmodell<sup>9</sup>.



Es zeigt die verschiedenen aufeinander aufbauenden Ebenen und Technologien, auf denen Semantic Web-Anwendungen basieren, von denen allerdings bisher nur die unteren vier Ebenen umgesetzt sind.

Die Kerntechniken URI, XML und RDF sind zum Teil unabhängig von der Idee des Semantic Web bzw. früher entwickelt worden. Grundsätzlich ist keine Technik, die dem Ziel dient, ausgeschlossen. Die Grafik macht auch deutlich, dass es sich bei dem Konzept um Evolution, nicht um Revolution handelt. Bereits existente Techniken werden nicht abgelöst, sondern erweitert und integriert.

Im Folgenden werden die einzelnen Ebenen des Schichtenmodells und ihre Funktionen eingehend erläutert.

### 2.1 Die Basistechnologien Unicode und URI

Die technologische Basis für das Semantic Web bilden Unicode und Uniform Resource Identifier (URI). Unicode stellt dabei die Kodierung aller Daten in einheitlichem Format sicher. Es handelt sich um einen internationalen Zeichensatz, der andere Zeichensätze wie z.B. den lateinischen, arabischen, kyrillischen oder japanischen, in sich vereint und daher für den Einsatz im WWW oder Semantic Web besonders geeignet ist.

<sup>7</sup> Mintert, Stefan: Das semantische Web. Abgehoben. In: iX-Special 1/2004, Seite 124

<sup>8</sup> <http://www.w3.org/2001/sw/>

<sup>9</sup> <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>

URI hingegen dient der eindeutigen Adressierung von Ressourcen. Es handelt sich um die verallgemeinerte Form des URL (Uniform Resource Locator), der Standard für die Bezeichnung von Web-Adressen ist. Ein maßgebliches Prinzip des Semantic Web ist, dass alles, was im Web über eine URI adressierbar ist, sich auch im Semantic Web beschreiben oder zur Beschreibung heranziehen lässt.

URIs sind auch ein erster Schritt der Beschreibung von Bedeutung. So kann beispielsweise die Stadt Paris durch den URI <http://www.paris.fr> beschrieben werden. Dies bietet für den menschlichen Nutzer die Möglichkeit, unter dieser Adresse nachzusehen und Informationen über diese Ressource zu beziehen. Allerdings liegt dort eventuell keine explizite Beschreibung der Bedeutung vor, sodass die Nutzung dem Menschen vorbehalten bleibt.

## 2.2 XML, XML-Schema und Namespaces

Die zweite Schicht ist durch die bereits standardisierten Sprachen XML und XML-Schema mit ihrem Konzept der Namespaces gegeben.

Ihre Funktion ist das Bereitstellen von Inhalt sowie die Kodierung und Serialisierung der Sprachen RDF/RDFS und OWL aus den höher liegenden Schichten. Der große Vorteil, den XML als Trägerformat bietet, ist seine leichte maschinelle Verarbeitbarkeit, die einfache Transformation in andere Formate sowie die unkomplizierte Einbettung in HTML-Dokumente.

Bei XML-Schema handelt es sich um eine Modellierungssprache, die XML um Datentypen erweitert sowie um Strukturierungs- und Wiederverwendungsmöglichkeiten und mit der sich Vokabulare definieren lassen. Allerdings wird hier für Dokumente lediglich eine rein syntaktische Festlegung vorgenommen, aber keinerlei Semantik beigefügt.

Mithilfe von Namespaces lassen sich Namenskonflikte bei Verwendung verschiedener XML-Sprachen in einem Dokument vermeiden.

So werden mittels folgender Namespaces die Präfixe `rdf`, `rdfs` und `owl` definiert, die im weiteren Dokument für eine eindeutige Verwendung sorgen.

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

```
xmlns:owl="http://www.w3.org/2002/07/owl#"
```

## 2.3 RDF und RDF-Schema

### 2.3.1 RDF

Das Resource Description Framework (RDF) stellt das grundlegende Datenmodell des Semantic Webs dar. Wie der Name sagt, kann man damit Gegenstände/Begriffe, so genannte Ressourcen, beschreiben, indem man Aussagen über sie aufstellt. Ziel ist hierbei die standardisierte Formulierung von Semantik, die Voraussetzung für die Nutzung durch Maschinen ist. RDF stellt den ersten wirklich neuen Standard im Schichtenmodell dar. Es ist ein recht offenes Datenmodell, das seine Aussagen in Form von dreigliedrigen Ausdrücken, auch RDF-Tripel genannt, formuliert.

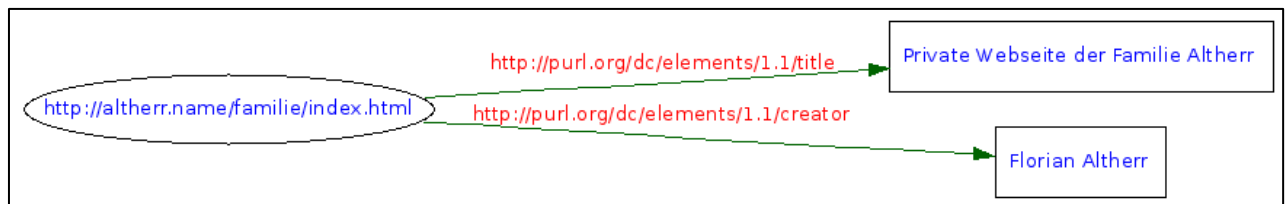
Sie bestehen aus Subjekt, Prädikat und Objekt, wobei das Subjekt eine Ressource ist, die durch die Eigenschaft (Prädikat) und deren Wert (Objekt)

beschrieben wird. Dadurch wird das Subjekt in Relation zum Objekt gesetzt. Es gibt verschiedene Darstellungsformen für ein Tripel:

So kann die Aussage über eine Webseite

<b>http://altherr.name/familie/index.html</b>	Subjekt
<b>ist erstellt von</b>	Prädikat
<b>Florian Altherr</b>	Objekt

- als Text formuliert werden: „http://altherr.name/familie/index.html ist erstellt von Florian Altherr“
- als Graph:



Die Beschriftung der Kanten stellt die Eigenschaft dar. In diesem Beispiel werden die Eigenschaften „title“ und „creator“ dem Vokabular des Dublin Core entliehen. Sie werden genau wie Subjekte und Objekte über ihren URI identifiziert. Nicht näher zu beschreibende Ressourcen werden Literale (Zeichenketten) genannt und im Graph als Rechtecke dargestellt.

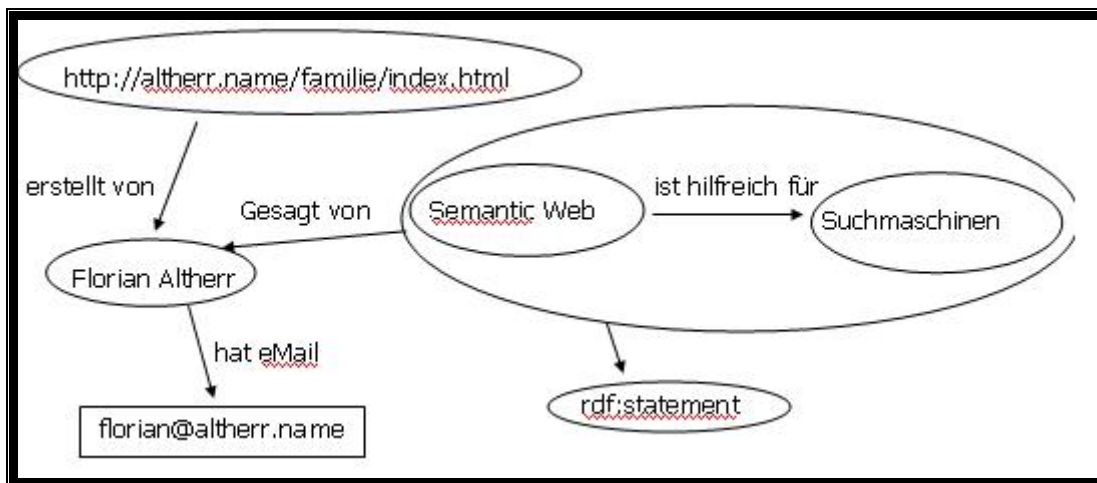
- in RDF/XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://altherr.name/familie/index.html">
    <dc:title>Private Webseite der Familie Altherr</dc:title>
    <dc:creator>Florian Altherr</dc:creator>
  </rdf:Description>
</rdf:RDF>
```

Die Eigenschaften der Ressource (Webseite) werden nicht bloß als Dokumente verlinkt, sondern durch Angabe der Prädikate semantisch in Beziehung gesetzt. Vorteil der Darstellung in RDF/XML ist die leichte Verarbeitbarkeit und Integrierbarkeit in andere W3C-Techniken.

Des Weiteren wird W3C-intern auch oft auf die Sprache N3 zurückgegriffen, die eine für Menschen besser lesbare Syntax aufweist, weil sie auf die XML-typischen Element- und Attributbezeichnungen verzichtet.

Es lassen sich auch Aussagen über Aussagen treffen. Das heißt eine Aussage wird selbst wieder Objekt einer anderen. Dies nennt man „zitieren“:



Außerdem gibt es die Möglichkeit sogenannte „blank nodes“ zu definieren. Darunter versteht man die Beschreibung einer unbenannten, „leeren“ Ressource, lediglich durch die Angabe ihrer Beziehungen zu anderen Ressourcen. Wichtig: RDF versucht nicht, alle möglichen Beziehungen inhaltlich zentral zu definieren, sondern stellt lediglich das Handwerkszeug zur Verfügung, mit dem jeder bereits kleine semantische Netze formulieren kann.

### 2.3.2 RDF-Schema

RDFS bedeutet Resource Description Framework Schema und ist wie RDF eine W3C-Empfehlung. Mit RDF lassen sich Informationen beschreiben und verknüpfen, was allerdings für ein tatsächliches semantisches Web noch nicht ausreicht. Die Bedeutung einer Ressource wird klarer durch die Beschreibung ihres Verhältnisses zu anderen Ressourcen. So lässt sie sich inhaltlich in Beziehung setzen und semantisch einordnen. Man könnte natürlich auf die Idee kommen jede einzelne Ressource dadurch zu definieren indem man ihr ihre Beziehungen zu anderen Ressourcen als RDF-Triple zuweist. Dass dies unnötig und uneffizient ist, lässt sich unschwer erkennen. Anstelle dessen ist es sinnvoller Ressourcen zu klassifizieren und diese Klassen untereinander zu verknüpfen und in Relation zu stellen („in Schubladen stecken“). Genau darin besteht die Erweiterung von RDF zu RDF-Schema. Nachdem Ressourcen in Klassen zusammengefasst sind, lassen sich einzelne Ressourcen als Erscheinungsformen, so genannte Instanzen, ihrer Ausgangsklassen beschreiben. In Abgrenzung zu XML, bei dem Dokumente sich nach Schemata oder DTDs einordnen lassen, bietet RDFS also über die syntaktische Klassifizierung hinaus die Möglichkeit Ressourcen inhaltlich (semantisch) miteinander in Beziehung zu setzen.

Konkrete Möglichkeiten zur Abbildung von Wissen in RDFS sind das bilden von Ressourcenklassen mit gemeinsamen Eigenschaften, das Aufstellen von Hierarchien (z.B. Subklassen) für Klassen und Eigenschaften, das Einschränken des Wertebereich oder der Domäne von Eigenschaften, die in den folgenden Beispielen verdeutlicht werden:

### Beispiel 1: subclassOf

```
<rdfs: Class rdf: ID="#Vater" >  
<rdfs: subclassOf rdfs:resource="#Mann"/>  
</rdfs: Class>
```

⇒ Unterklasse: Jede Instanz der Klasse Vater ist eine Instanz der Klasse Mann

### Beispiel 2: subProperty, range, domain

```
<rdf: Property rdf: ID="#vaterVon" >  
<rdfs: subPropertyOf rdf: resource="elterVon"/>  
<rdfs: domain rdf: resource="#Vater"/>  
<rdfs: range rfd: resource="#Mensch"/>  
</rdf: Property>
```

⇒ Eigenschaft vaterVon wird definiert als Untereigenschaft von elterVon, zusätzlich werden Bild- und Wertebereich der Eigenschaft eingeschränkt auf „Mensch“ und „Vater“.

Mittels RDFS lassen sich Taxonomien erstellen. Sie sind eine Art "kleiner Bruder" von Ontologien und ergeben ein Bild oder Modell eines Wissensgebietes. Allerdings ermöglichen Ontologien noch genauere Beschreibungen der Beziehungen von Ressourcen und deren Eigenschaften zueinander.

## **2.4 Ontologien, OWL und seine Vorgänger**

Ursprünglich stammt der Begriff „Ontologie“ aus der Philosophie, wo er eine "Seinslehre" bezeichnet. Im Rahmen des Semantic Web wurde er übernommen und definiert als Modellierung von einem Teilbereich (Domän) der realen Welt mit dem Ziel des Aufbaus einer strukturierten Wissensbasis.

Man versteht also unter Ontologien Modelle, die Klassen von RDF-Ressourcen, ihre Eigenschaften und deren Beziehungen zueinander definieren und zwar, abhängig von der Granularität der verwendeten Ontologiesprache, noch wesentlich detaillierter als in RDFS möglich. Mit der Zeit wurden diverse Ontologiesprachen entwickelt, wobei OWL – die Web Ontology Language – seit 2004 der vom W3C empfohlene Standard zur Realisierung des Semantic Webs ist. Sie verfügt über eine in Beschreibungslogik formulierte Semantik und erlaubt Aussagen, die Beziehungen zwischen Konzepten und logische Schlussregeln für sie festlegen.

Das Fundament von OWL bilden RDF und RDFS, sie sind Untermengen der sprachlichen Ausdrucksfähigkeit.

Frühe Ontologiesprachen waren das bereits zu Beginn erwähnte SHOE (Simple HTML Ontology Extension), sowie DAML+OIL, das aus den Projekten des Defense Advanced Research Projects Agency (DARPA) und dem Ontology Inference Layer hervorging. Für die zukünftige Arbeit soll allerdings die Energie der verschiedenen Forschungsgruppen auf OWL konzentriert werden, das in 3 Varianten, unterschieden nach ihrer Komplexität/Mächtigkeit daher kommt:

- OWL-Lite: Die ausdrücksschwächste, sie bietet kaum mehr Möglichkeiten als RDFS, es lassen sich nur Klassenhierarchien und einfache Rollenbeschränkungen definieren

- OWL-DL (Description Logics): Die für den praktischen Einsatz geeignetste (Verhältnis Komplexität - Entscheidbarkeit gewahrt)
- OWL-Full: ausdrückstärkste, stellt zahlreiche Sprachelemente zur Verfügung, die Verhältnisse zwischen Klassen und ihren Eigenschaften sehr differenziert beschreiben. Bietet aber keine Garantie zur Berechenbarkeit (nicht entscheidbare Sprache). Für den praktischen Einsatz daher eher ungeeignet.

Neben den bereits erläuterten Möglichkeiten der Bildung von Klassen, Unterklassen, Eigenschaften, Untereigenschaften, Instanzen und Einschränkungen von Wertebereichen und Domänen, bietet OWL erweiterte Beschreibungsmöglichkeiten für

- Eigenschaftsrestriktionen
- Schnittmenge, Vereinigung und Komplement
- Kardinalitäten (minimale, maximale und exakte)
- Äquivalenzen
- Charaktere von Eigenschaften: symmetrisch, transitiv, funktional

### 2.4.1 Syntaxbeispiele

Klassen betreffend

- `<owl:class>`
- `<owl:unionOf>`
- `<owl:intersectionOf>`
- `<owl:equivalentOf>`

Eigenschaften betreffend

- `<owl:Restriction>`
- `<owl:allValuesFrom>`
- `<owl:someValuesFrom>`

### 2.4.2 Umfassendes Beispiel: Bürgeramt.owl

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:ba="http://altherr.name/Buergeramt.owl#">

  <owl:Class rdf:ID="Person"/>

  <owl:Class rdf:ID="Geschlecht"/>

  <owl:Class rdf:ID="Frau">
    <rdfs:subClassOf rdf:resource="#Person"/>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#geschlecht"/>
        <owl:hasValue rdf:resource="#weiblich" rdf:type="#Geschlecht"/>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
```

```

<owl:ObjectProperty rdf:ID="geschlecht"
  rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
  <rdfs:range rdf:resource="#Geschlecht"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="name"
  rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="vorname"
  rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:DatatypeProperty>

<ba:Person rdf:ID="LAltherr"
  ba:vorname="Lena"
  ba:name="Altherr">
  <ba:geschlecht rdf:resource="#weiblich"/>
</ba:Person>

</rdf:RDF>

```

Das Beispiel beschreibt die Konzepte <Person>, <Geschlecht> und <Frau>. Eine Frau ist definiert, als eine <Person> mit dem Wert <weiblich> im Property <geschlecht>, welches wiederum der Klasse <Geschlecht> angehören muss. Die Instanz <LAltherr> ist somit als <Person> beschrieben eine Frau (<Frau>). Mittels Inferenz, dem logischen Schlussfolgern aus Ontologien, kann diese Zugehörigkeit ermittelt werden.

## 2.5 Logic, Proof, Trust

Die bisher beschriebenen Daten- und Metadatenstandards sorgen für die wohldefinierte Bedeutung - die Voraussetzung für die Umsetzung unserer Vision, das Web für Maschinen/Agenten verständlich zu machen.

Um nun aus dem abgebildeten Wissen Schlüsse ziehen zu können und gegebenenfalls neues Wissen zu bilden, sind weitere Schritte notwendig, die in der **Logikebene (logic layer)** anzusiedeln sind. Zum einen ist es erforderlich, Regeln nach dem WENN-DANN-Prinzip aufzustellen. Dies umfasst nicht nur Schlussfolgerungen, die sich implizit aus der Definition Daten mit RDF(S) ergeben, wie z.B. die hierarchische Gliederung von Subklassen (subClassOf), sondern auch je nach Belieben komplexe Regeln.

Zum andern werden Programme benötigt, die mittels dieser Regeln aus dem abgebildeten Wissen logisch schließen können. Diese Programme werden „inference engines“ genannt und dienen 2 maßgeblichen Zwecken: Der Instanzprüfung, dem „Instance checking“, worunter man das RDF-Gegenstück zur XML-Validierung versteht. Wieweit eine Instanzprüfung erfolgen soll und wie tolerant mit den RDF-Daten umgegangen werden soll, müssen Entwickler selbst entscheiden. Der zweite Zweck ist das Schema oder ontology mapping, welches der Angleichung von Schemata und Ontologien dient. Dieses ist genauso wichtig wie die Instanzprüfung, denn eine Anwendung müsste zusätzlich zu Abfragesprachen auch logische Regeln interpretieren können, und so Parallelitäten oder Unterschiede erkennen. Um beim Beispiel der Weinontologie

(wo) zu bleiben: sie müsste den Zusammenhang der Aussagen „wo:hat\_gekeltert“ und „ao:gekeltert\_von“ einer anderen Ontologie (ao) erkennen.

Die nächste Ebene, **Beweisebene (proof layer)** versucht nun umgekehrt nach den gleichen logischen Regeln die logische Wahrheit bestimmter RDF-Aussagen zu prüfen, indem sie untersucht, ob diese aus anderen Aussagen herzuleiten sind. Die Regeln und Daten, mit denen diese Ebene arbeitet, entsprechen denen der Logikebene, daher werden beide auch oft zusammen erwähnt. Die auswertenden Programme allerdings funktionieren anders und müssen in speziellen Sprachen (proof languages) geschrieben werden. In diesem Bereich sind bisher erst Ansätze zu erkennen, es existieren keine konkreten Anwendungen.

Da das RDF-Datenmodell von Haus aus dezentral und offen ist und somit jeder beliebige Aussagen über beliebige Ressourcen treffen kann, kommt der **Vertrauensebene (trust layer)** des Semantic Webs eine besondere Bedeutung zu.

Sie soll die Gefahr inhaltlich falscher Aussagen eindämmen, indem sie bei der Bildung von Wissen nur vertrauenswürdige Quellen berücksichtigt. Dabei favorisiert das W3C bei der Umsetzung vor allem technische Standards der Verschlüsselung, digitale Signaturen (sogenannte XML Signatures). Ein weiteres Konzept sind trust-network-Verfahren, bei denen für Quellen verschiedene Vertrauenslevel zugewiesen werden. Vertrauen wird praktisch über Zwischenstationen im Netzwerk hergestellt. (A vertraut B, B vertraut C => A vertraut C in gewissem Maße). Das Vertrauen sinkt allerdings je mehr solcher Zwischenstationen es gibt. Auch in diesem für das Durchsetzen von Semantic Web-Technologien wichtigen Bereich existieren bisher bis auf die erwähnten Techniken kaum Ansätze.

### 3. Anwendungsbeispiele

Bisher sind praktisch einsetzbare Anwendungen mit Semantic Web Technologien noch rar. Jedoch werden teilweise einzelne Techniken aus dem Schichtenmodell eingesetzt: So benutzen RSS in der Version 1.0 und Mozilla RDF zum Speichern von Webseiteninhalten bzw. zum Ablegen von Konfigurationsdaten und Bookmarks. Von Fujits und Ricoh gibt es eine Software (OKAR), die es in Firmennetzen erlaubt eine bestimmte Person mit bestimmten Fähigkeiten zu suchen ([www.labs.fujitsu.com/en/techinfo/okar/](http://www.labs.fujitsu.com/en/techinfo/okar/)). Am Fraunhofer Institut wird mit dem MPEG-7 Standard versucht die Annotierung von Multimedia umzusetzen. Adobe stellt das Tool XMP ([www.adobe.com/products/xmp](http://www.adobe.com/products/xmp)) zur Verfügung, mit dessen Hilfe man fast alle Adobe-Formate mit semantischer Information anreichern kann.

Daneben existiert mittlerweile eine kleine Schar von Ontologieeditoren, wie Protégé von der Stanford University oder Semantic Works 2006 von Altova, die das Erstellen von Ontologien erleichtern sollen. Dies führte bereits zu einer größeren Menge von detaillierten Ontologien für verschiedene Themenbereiche des Alltags oder spezieller Anwendungsgebiete. Hier sind besonders die beiden Ontologiesammlungen [www.geneontology.org](http://www.geneontology.org) und [www.schemaweb.info](http://www.schemaweb.info) zu nennen.

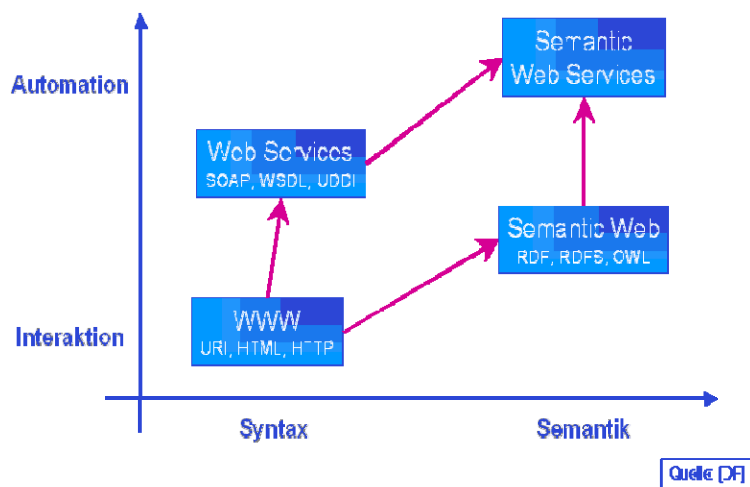
Mit OntoWeb ([www.ontoweb.org](http://www.ontoweb.org)) wurde der Versuch gestartet, ein ganzes Portal auf Basis semantischer Techniken aufzubauen. Es erlaubt in Zukunft z.B. auch

das automatisierte Einloggen anderer Portale und das Einstellen von deren auch wiederum semantisch ausgezeichneten Inhalten.

Ein recht anschauliches Projekt stellt das Netzwerk von Friend of a friend ([www.foaf-project.org/](http://www.foaf-project.org/)) dar, das sich besonders schön mit dem Firefox-Plugin [Piggy Bank](#) betrachten lässt. Hier werden RDF-Technik und Ontologien gemeinsam benutzt. Die zentrale Rolle kommt hierbei dem Tag `<foaf:knows>` zu, durch das, das eigentliche soziale Netzwerk zustande kommt. An diesem Tag können sich Anwendungen „entlanghangeln“ und das Netzwerk grafisch darstellen. Siehe dazu <http://www.students.uni-mainz.de/faltherr/foaf.rdf> (mit Piggy Bank).

### 3.1 Semantic Web Services

Unter Semantic Web Services versteht man die Erweiterung des Web Service-Konzepts durch Techniken des Semantic Webs. Sie stellen ein Fernziel bei der Forschung im Bereich Web Service und Semantic Web dar. Dabei soll es Agenten möglich gemacht werden mit Hilfe von Service-Ontologien Semantic Web Services aufzufinden, aufzurufen, zusammen zu stellen und zu überwachen. Durch die Komposition von Semantic Web Services sollen komplexe Verhaltensweisen ermöglicht werden.



10

Die Grafik zeigt den Stand der Dinge, wo sich Forschung und Praxis derzeit befinden und wo sie hinwollen. Das WWW mit seinen Techniken URI, HTML, und http hat sich etabliert. Es wird deutlich, dass dies nicht das Ende der Entwicklung sein soll. Es geht zum einen darum, sich von der Interaktionsebene hin zu mehr Automation zu entwickeln. Zum anderen ist eine Weiterentwicklung von der Syntax- auf die Semantikebene angestrebt. Die unterschiedlichen Forschungsgebiete (Web Services und Semantic Web) und ihre Technologien, die aus den Möglichkeiten des WWW entstanden sind und deren Erweiterungen darstellen, sollen schließlich im Komplex der Semantic Web Services zusammenfließen.

<sup>10</sup> Quelle: <http://www.jeckle.de/semanticWebServices/intro.html>

Neben der Arbeit mit SOAP und dem Verzeichnisdienst UDDI ist ein Teil der Arbeit in diesem Bereich die Beschreibung von Web Services, die man gerne um Semantic Web Technologien erweitern würde. Hatte man bisher auf die Verwendung von **WSDL**, der Web Service Description Language, gesetzt, so sind deren Nachteile deutlich sichtbar: Ein WSDL-Dokument enthält funktionelle Angaben über die Schnittstelle (Interface) des Services, Zugangsprotokoll und Details zum Deployment, sowie alle notwendigen Informationen zum Zugriff auf den Service in einem maschinenlesbaren Format. Woran es WSDL aber mangelt, sind Quality of Service-Informationen, und die Möglichkeit mit Taxonomien/Ontologien einen Service semantisch einzuordnen.

Hier kommt die neue Sprache **OWL-S** ins Spiel, die auf der Basis von OWL das Auffinden, Ausführen und Komponieren von Web Services ermöglichen soll.

Sie lässt viel Raum für weitergehende Konzepte und ist durch 3 Grundbestandteile charakterisiert: Profile (beschreiben was ein WS leistet), Prozessmodelle (beschreiben wie er dies tut) und Groundings (beschreiben die Möglichkeiten des Zugriffs auf den WS).

OWL-S bietet also den Vorteil einer eindeutigen, formal spezifizierten Semantik von WS-Parametern durch Ontologien und eines einfachen Integrierens anderer Datenstrukturen.

Bis zur Realisierung von wirklichen Semantic Web Services, die hochkomplexe Aufgaben übernehmen können, wartet noch viel Arbeit auf Forscher und Entwickler, allerdings auch ein faszinierendes und viel versprechendes Forschungsgebiet.

#### **4. Fazit und Ausblick**

Aus Sicht des W3C ist die erste wichtige Phase, die "Semantic Web Activity Phase 1" vollendet: Die Kerntechnologien und Infrastruktur sind implementiert, RDF und OWL spezifiziert, zum Teil wurden auch Ontologien geschaffen, die viele Ausschnitte des Lebens abbilden können. Das Semantic Web existiert also in kleinen Ansätzen. In der zweiten Phase nun gilt es die Verbreitung und Anwendung von Techniken aus dem Schichtenmodell des Semantic Webs voranzutreiben. Dazu müssen vermehrt einfache Werkzeuge wie Editoren und Abfragesysteme bereitgestellt werden, die Autoren und Entwicklern das Arbeiten mit diesen Techniken erleichtern, denn momentan ist das Annotieren von Daten eine mühsame Angelegenheit und mit ein Grund für die noch geringe Zahl praxistauglicher Anwendungen.

Positiv ist die allgemeine Akzeptanz, dass ein Semantic Web mit all seinen teils unvorstellbaren Möglichkeiten existieren sollte, allerdings ist der Weg dorthin noch weit. Da bisher keine „Killer Application“ entdeckt wurde, bleibt nur der beschwerliche Weg über die Bottom-Up Entwicklung, vorangetrieben von Anwendern. So erhofft man sich die Ausweitung der Entwicklung aus speziellen Anwendungsdomänen, wie Firmennetzwerken hin zur allgemeinen Verfügbarkeit. Dazu gehört in besonderem Maße auch die Motivation neuer Anbieter und Nutzer zur Mitarbeit.

Bereiche mit Entwicklungsbedarf, in denen man sich aber wegen der Vielzahl an potentiellen Nutzern und Nutzungsmöglichkeiten auch große Erfolge verspricht, sind die Integration von Multimedia im Semantic Web (beispielsweise Anreicherung von Musikdateien mit semantischer Information) bzw. die Anbindung an andere Datenquellen wie Relationale Datenbanken, die schon Semantik enthalten.

Von Kritikern der Semantic Web-Idee werden auch Risiken vorgetragen, so zum Beispiel die sinkende Ergebnisbreite durch die automatische Nutzung semantischer Techniken bei der Suche. Sie eliminiert alle Zufallsfunde, die uns Menschen nach dem Prinzip „nicht gefunden aber dennoch interessant“ begegnen. Negativ bewertet wird auch die sinkende Vertraulichkeit: Darunter versteht man die Tatsache, dass durch die eindeutige Referenzierung die Anonymität von Datenquellen und Autoren leidet. Schließlich verführe ihrer Meinung nach die Verlagerung semantischer Filterungsprozesse an Maschinen zu absichtlich falschen Beschreibungen, das heißt zu sinkender Verlässlichkeit von Informationen. Auch bei der Modellierung von abstrakten Begriffen wie „Freiheit“ oder „Liebe“ tut man sich bisweilen noch schwer.

Eine besonders kritische Stellungnahme, die hier der Vollständigkeit halber noch erwähnt sein soll, kommt vom Google-Mitbegründer Sergey Brin, der sagte: "I'd rather make progress by having computers understand what humans write, than by forcing humans to write in ways computers can understand"<sup>11</sup> und damit für den Fortschritt in der Künstlichen Intelligenz eintritt statt für die Problemlösung durch Wohldefiniertheit der Daten und der Operationen auf ihnen, wie es das Semantic Web vorsieht.

## 5. Quellenverzeichnis

### Literatur:

Berners-Lee, Tim, Hendler, James und Lassila, Ora: The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, Scientific American, May 17, 2001

Mintert, Stefan: Das semantische Web. Abgehoben. In: iX-Special 1/2004, Seite 123-125

Ziegler, Cai: Surfende Maschinen. Das XML-Kompendium. In: iX-Special 1/2004, Seite 126-130

### WWW-Quellen:

Altova: Was ist das Semantic Web? Zitiert nach [http://www.altova.com/de/semantic\\_web.html](http://www.altova.com/de/semantic_web.html)

Behrendt, Wernher: Semantisches Web – Das Netz der Bedeutungen im Netz der Dokumente. Zitiert nach: [http://www.big.tuwien.ac.at/teaching/offer/ss05/we\\_vo/kapitel14.pdf](http://www.big.tuwien.ac.at/teaching/offer/ss05/we_vo/kapitel14.pdf)

Berners-Lee, Tim/ Miller, Eric: Das semantische Web hebt ab. Zitiert nach <http://www.w3c.de/Misc/tbl-en-sw.html>

---

<sup>11</sup> Zitiert nach: Matthias Brake: Das semantische Web. In Telepolis vom 12.12.2005 <http://www.heise.de/tp/r4/artikel/21/21498/1.html>

Dostal, Wolfgang/ Jeckle, Mario / Melzer, Ingo u.a. : Semantic Web. In: Objekte 5/2004. Zitiert nach:  
[http://www.sigs.de/publications/os/2004/05/dostal\\_melzer\\_OS\\_05\\_04.pdf](http://www.sigs.de/publications/os/2004/05/dostal_melzer_OS_05_04.pdf)

European Research Consortium for Informatics and Mathematics. Number 51 October 2002, Special Theme: Semantic Web. Zitiert nach:  
[http://www.ercim.org/publication/Ercim\\_News/enw51/EN51.pdf](http://www.ercim.org/publication/Ercim_News/enw51/EN51.pdf)

Jeckle, Mario/Dostal, Wolfgang: Semantik, Odem einer Service-orientierten Architektur. In Java Spektrum 1/2004, Seiten 53-56. Zitiert nach:  
<http://www.jeckle.de/semanticWebServices/intro.html>  
Notholt, Jochen: Die Standards des Semantic Web (Teil 2) 2005  
<http://www.jurpc.de/aufsatz/20050065.htm>, Stand 10.02.2006

Tolksdorf, Robert: Projekt Semantic Web (WS 2005/2006) Zitiert nach:  
[http://www.inf.fu-berlin.de/inst/ag-nbi/lehre/0506/P\\_SW/fohlen/Einleitung.pdf](http://www.inf.fu-berlin.de/inst/ag-nbi/lehre/0506/P_SW/fohlen/Einleitung.pdf)

W3C, OWL Web Ontology Language Overview: Deutsche Übersetzung  
Köstlbacher, Anton: <http://www.semaweb.org/dokumente/w3/TR/2004/REC-owl-features-20040210-DE.html>

Weitere:

<http://www.w3.org/2001/sw/>  
<http://www.w3.org/People/Berners-Lee/>  
<http://www.w3.org/DesignIssues/Semantic.html>  
<http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>

[http://de.wikipedia.org/wiki/Semantisches\\_Web](http://de.wikipedia.org/wiki/Semantisches_Web)  
[http://de.wikipedia.org/wiki/Web\\_Ontology\\_Language](http://de.wikipedia.org/wiki/Web_Ontology_Language)  
[http://de.wikipedia.org/wiki/Resource\\_Description\\_Framework](http://de.wikipedia.org/wiki/Resource_Description_Framework)

<http://images.google.com/images?client=opera&rls=en&q=Jaguar&sourceid=opera&ie=utf-8&oe=utf-8&sa=N&tab=wi>