

# Semantic Web im Überblick

Johannes Vetter  
vetterj@ifi.lmu.de

Universität München  
Amalienstrasse 17, 80333 München, Deutschland

**Zusammenfassung** In der folgenden Ausführung geht es darum einen Überblick über das Semantic Web zu schaffen. Zuerst wird beschrieben wie sich das Semantic Web entwickelt hat. Dabei wird sowohl über ältere Technologien wie HTML als auch über XML gesprochen.

Des Weiteren wird eine allgemein gültige Definition für den Begriff des Semantic Webs eingeführt. Darauf aufbauend werden dann Technologien erklärt die für das Semantic Web nötig sind.

Insbesondere wird hierbei auf den so genannten "Stack of Languages" und den dort enthaltenen Technologien wie RDF, OWL und URI eingegangen.

Es wird die Bedeutung von Ontologien betrachtet und am Schluss wird anhand von Beispielen ein Ausblick gegeben, für welche Anwendungen das Semantic Web nützlich sein kann.

## 1 Einleitung

Seit der Erfindung und erstmaligen Einführung des World Wide Webs im Jahre 1989 durch eine Forschungsgruppe um den Erfinder Sir Tim Berners Lee hat sich das Hypertext System WWW rasant entwickelt.

Gerade in der Art der Anwendung und in der Quantität der Nutzung haben sich erhebliche Änderungen ergeben. Von einer Technologie die zum einfachen Austausch von Forschungsarbeiten gedacht war wurde schnell eine weltweit genutzte Plattform, die bald neben Text auch Bilder, Musik und andere dynamische Inhalte darstellen konnte. Die schnelle Verbreitung ist vor allem auf die Philosophie zurückzuführen nur patentfreie Standards zu verwenden.

Die schnelle Ausbreitung des World Wide Web führte dazu, dass immer mehr Menschen Zugang zu diesem Medium bekamen und dadurch die Möglichkeit erhielten Inhalte im Internet bereit zu stellen. Die freie Zugänglichkeit führte zu einer immensen Menge an Informationen, die nicht mehr zu überschauen ist und Tag für Tag ansteigt. Neueste Studien von Netcraft <http://news.netcraft.com/> sprechen von mehr als 60 Millionen Servern weltweit. Deshalb ist es essentiell, dass es möglich wird die bereitgestellten Inhalte durch Maschinen zu durchsuchen um an die gewünschten Informationen zu gelangen. Das Problem hierbei ist jedoch, dass es bei der Freigabe von Inhalten momentan keine semantischen Vorgaben gibt, was dazu führt, dass Inhalte zwar für den Menschen lesbar und auch verständlich sind, Maschinen jedoch nur die Möglichkeit haben nach bestimmten Algorithmen und Heuristiken in Dokumenten zu suchen. Das bedeutet die Inhalte können von Maschinen nicht verstanden und interpretiert werden, was die Suche ungenau und verfälschbar macht.

Genau diesem Missstand soll durch die Einführung semantischer Grundlagen entgegengetreten werden.

## 2 Semantic Web Überblick

Aus den eben geschilderten Problemen hat sich Handlungsbedarf ergeben. Es war wieder Sir Tim Berners Lee, der in diesem Fall die Initiative ergriffen hat. Das W3 Consortium treibt seit Ende der 1990er Jahre die Entwicklung des "Semantic Web" voran. Und obwohl diese Technologie noch in ihren Anfängen ist, gibt es schon zahlreiche Publikationen und Werkzeuge die darauf aufbauen.

### 2.1 Begriffsklärung und Definition des Semantic Web

Um über das Semantic Web reden zu können ist es zu allererst wichtig zu verstehen, was Semantik überhaupt bedeutet.

Man spricht in der Kommunikation von drei Bausteinen, die Syntax, die Semantik und die Pragmatik. Die Syntaktik beschränkt sich auf die Bildung von gültigen Zeichenketten und stellt damit die unterste Ebene der Kommunikation dar.

Die Semantik hingegen bildet die mittlere Schicht und untersucht die Beziehung zwischen den Zeichen und den Objekten. Jedes Zeichen hat hierbei eine Bedeutung und bezeichnet ein Objekt.

In der obersten Schicht, der Pragmatik, wird die Anwendung und Wirkung von Zeichen untersucht.[1]

Bei Semantic Web handelt es sich nicht um eine Neuentwicklung, sondern um die Erweiterung des schon vorhandenen World Wide Webs.

Das Problem am heutigen World Wide Web ist, dass nur die Mensch zu Mensch Kommunikation implementiert ist. Das bedeutet, dass nur Menschen in der Lage sind die Inhalte die von anderen Menschen bereitgestellt wurden vollständig zu verstehen und zu interpretieren.

Eine viel wichtigere Kommunikationsrichtung, die es erst möglich macht maschinelle Ansätze zu implementieren ist jedoch die Mensch-Maschine und Maschine-Maschine Kommunikation. Diese sind bis jetzt noch nicht ausgereift und sollen durch das Semantic Web möglich werden.

Der Grundgedanke ist es die bereitgestellten Informationen nicht nur in einer für den Menschen lesbaren, sondern zusätzlich auch in einer für Maschinen verarbeitbaren Form zu repräsentieren. Das bedeutet, die Daten bekommen eine wohl definierte Bedeutung die durch den Menschen hinzugefügt wird. Die Daten werden also explizit um Aussagen über ihre Bedeutung, sprich Semantik angereichert.

Es soll dadurch die Möglichkeit geschaffen werden Aktivitäten, die momentan nur von Menschen durchgeführt werden können auch auf Maschinen zu übertragen, da diese schneller und effizienter Arbeiten können, wenn die entsprechenden Voraussetzungen gegeben sind. Dadurch wird der Mensch bei der Nutzung des Webs unterstützt.

Des Weiteren sollen die Informationen über einen eindeutigen Bezeichner identifiziert werden können. Die so genannten URIs (Uniform Resource Identifiers) dienen dazu den Daten ein Anwendungsgebiet zuzuweisen, für welches sie relevant sind. Das ist notwendig um Doppeldeutigkeiten in Begriffen auszuschließen.

So ist zum Beispiel die Zahl 100 in einer Vielzahl von semantischen Abhängigkeiten denkbar. Es ist nun notwendig zu bezeichnen ob es sich zum Beispiel um 100EUR oder 100Km/h handelt.

Generell lässt sich sagen, dass das Semantic Web folgende drei Probleme lösen muss:

- Homonyme: Dasselbe Wort kommt in unterschiedlichen Kontexten mit unterschiedlicher Bedeutung vor
- Synonyme: Dasselbe Konzept wird in unterschiedlichen Kontexten unterschiedlich benannt
- Prioritäten: Inhalte müssen nach ihrer Wichtigkeit zu einem Bestimmten Kontext geordnet werden können [2]

Um diese Probleme in den Griff zu bekommen baut das Semantic Web auf einer Reihe von Technologien und Standards auf die in den nächsten Abschnitten beschrieben werden. Um die folgenden Ausführungen besser zu verstehen folgt nun eine allgemeine Definition des Semantic Web, wie sie von der "Semantic Web Agreement Group (SWAG)" aufgestellt wurde: [3]

Das Semantic Web ist ein Netz welches eine Menge von Dokumenten beinhaltet. Des Weiteren beschreibt es explizite Abhängigkeiten zwischen Teilen des Semantic Webs und beinhaltet semantische Informationen die für automatisierte Vorgänge von Maschinen vorgesehen sind. [4]

## **2.2 Motivation zur Entwicklung des Semantic Web**

Als sich das World Wide Web entwickelt hat basierte es ausschließlich auf Markup Sprachen wie HTML. Diese Sprachen sind dazu geeignet Inhalte in einem gewissen Layout darzustellen. Computer haben in dieser Zeit eine eher passive Rolle gespielt und waren prinzipiell nur für die Darstellung der Inhalte im vorgegebenen Layout zuständig. Man könnte Computer in dieser Funktion mit Fernsehern vergleichen. Wie schon angesprochen ist es für Maschinen nicht möglich den Inhalt zu verstehen.

Nach einiger Zeit entwickelte sich das World Wide Web jedoch weiter und es wurden Services angeboten die nach einer Markupsprache verlangten, die auch für Maschinen lesbar ist. So war es beispielsweise im E-Commerce Sektor von entscheidender Bedeutung, dass Computer nicht nur das Layout einer Seite anzeigen konnten, sondern dass sie auch mit Inhalten wie "item price" und "sale offer" umgehen konnten.

Dafür war es also nötig eine Markupsprache zu finden die über HTML hinaus geht. Eine Sprache die dafür geeignet ist gab es schon. SGML konnte als Grundlage dienen. Es wurde durch Komplexitätsreduktion in XML überführt, was dann für die angesprochenen Zwecke verwendet werden konnte. HTML wird jedoch immer noch für das Layout der Inhalte benötigt, allerdings in der abgewandelten Form XHTML.

Wie sich zeigt, ist XML der erste Versuch gewesen Semantik in Webinhalte zu bringen. Eine weiterer Schritt in diese Richtung ist der Dublin Core Standard. Dies ist ebenfalls eine Möglichkeit vor allem Webseiten mit Semantik auszustatten.

Hinter dieser Idee verbirgt sich die Eingabe von Metadaten zu speziell standardisierten Elementen. Durch diese Technik kann beispielsweise der Autor ("creator") einer Webseite so angegeben werden, dass es für eine stichwortbasierte Suchmaschine möglich ist diesen auszulesen.

Trotz dieser Möglichkeiten Semantik einzubetten sind Technologien und Standards wie XML und Dublin Core nicht stark genug um das Semantic Web darauf zu errichten, jedoch macht es die zugrunde liegende XML Technologie möglich weitere Schritte in Richtung Semantic Web zu machen.

Im folgenden Abschnitt wird erläutert welche Technologien sich aus XML entwickelt haben beziehungsweise welche Technologien des Semantic Web XML verwenden und wie diese den "Stack of Languages" für das Semantic Web bilden.

Zum Schluss dieser Themenmotivation soll nun noch angegeben werden was Semantic Web nicht ist, da hier oft falsche Vorstellungen aufkommen.

Beim Semantic Web handelt es sich nicht um eine hoch ausgereifte künstliche Intelligenz, die in der Lage ist menschliche Sprache zu verstehen. Vielmehr ist es die Fähigkeit wohldefinierte Probleme durch wohldefinierte Operationen auf wohldefinierten Daten zu lösen. Das bedeutet, es muss zuerst extra Arbeit investiert werden um später davon zu profitieren. [4] [5]

### 2.3 Standards und zugrunde liegende Technologie

Wie schon erwähnt bildet das Semantic Web eine Erweiterung schon existierender Technologien und Standards. Demzufolge lässt sich die Architektur des Semantic Web in Ebenen wie in Abbildung 1 am besten veranschaulichen.

Im Folgenden werden die einzelnen Ebenen kurz erwähnt. Detailliert werden die einzelnen Technologien anschließend getrennt behandelt.

**Semantic Web Architektur** Die Basis der Architektur bilden bereits bekannte Ebenen. Die Neuerungen bauen darauf auf.

Die unterste Ebene wird durch URIs und Unicode Technologie gebildet. URIs sind dazu da um Ressourcen im Web eindeutig identifizieren zu können. Zusammen mit dem XML Namespace kann jeder seine eigenen Ressourcen benennen ohne dass die Gefahr von Namenskollisionen besteht. [4] [5]

Der Unicode Standard ist entscheidend um die Benennung von Ressourcen in jeder erdenklichen Sprache möglich zu machen. Dadurch wird ein multinationaler Austausch möglich.

Da XML wie schon angesprochen die Basistechnologie bildet wird es auch in dieser Architektur als zweite Ebene eingeführt. XML dient zur Speicherung und dem universellen Austausch von Daten. Darauf können nun die anderen Technologien aufsetzen.

Die dritte Schicht ist die wohl tragendste von allen. RDF ermöglicht es Metadaten in maschinenlesbarer Form zu erstellen. Dabei wird eine Art Satzbau verwendet auf den später näher eingegangen wird.

Ebene vier sind die Ontologien die ebenso wie das RDF einen wichtiger Pfeiler für das Semantic Web darstellen. Sie dienen zur Kommunikation verschiedener,

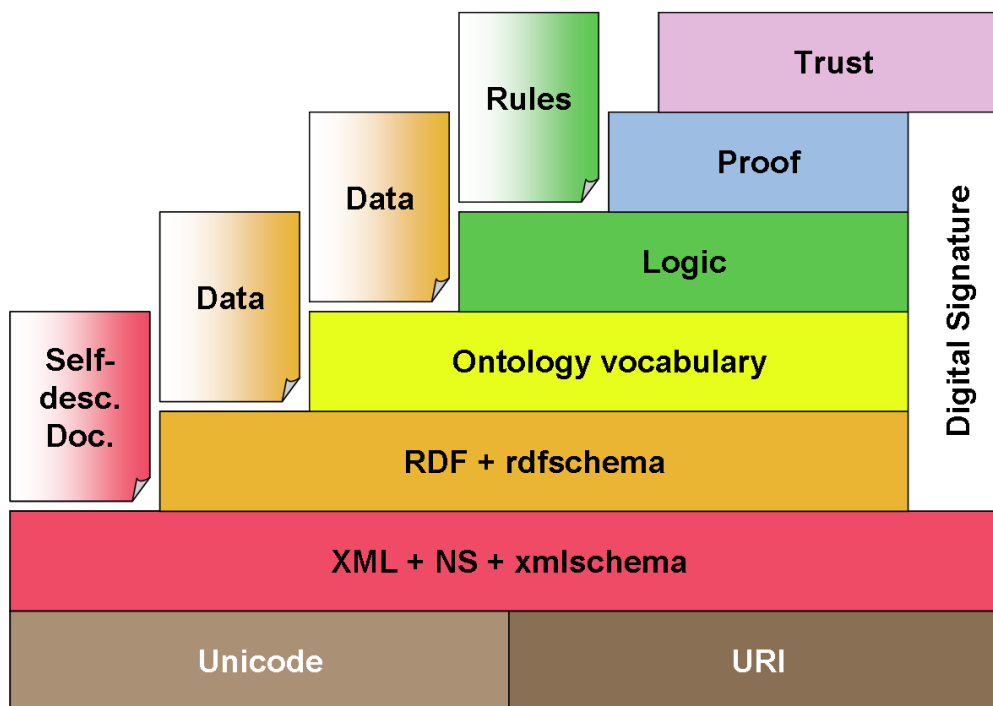


Abbildung 1. Semantic Web Architekturstack [5]

voneinander unabhängigen Domänen auf semantischer Ebene. Hier können Probleme auftreten die schon in der Begriffserklärung erläutert wurden. So kann es dazu kommen, dass dieselbe Ressource über verschiedene URIs identifiziert ist, ein so genanntes Synonyma. Diese Probleme können durch Ontologien gelöst werden.

Die obersten Schichten der Architektur werden gebildet durch Logic, Trust und Proof. Die Logic Ebene soll es Computern durch Regeln ermöglichen Muster zu erkennen um neues Wissen zu erlangen.

Die Proof Ebene soll es möglich machen die Vertrauenswürdigkeit von Ressourcen zu unterscheiden. Zusammen mit der Digital Signature Schicht soll es möglich sein, dass Computer vertrauensvolle Aufgaben bewältigen können, ohne dass ein Mensch etwas dazu tun muss.

Im Folgenden werden die einzelnen Bausteine im Detail behandelt.

**URIs** Um Ressourcen im Web identifizieren zu können ist es nötig über URIs zu verfügen. Der Name Uniform Resource Identifier leitet sich ab aus der Tatsache, dass ein einheitliches System zugrunde liegt und jede ansprechbare Ressource eine URI besitzt. Prinzipiell kann man allem und jedem eine URI geben. Sobald etwas eine URI besitzt spricht man davon, dass es im Web ist.

Das bekannteste Beispiel einer URI ist wohl die URL (Uniform Resource Locator). Wenn man die URL entsprechend zerlegt, wird dem Computer dadurch mitgeteilt welche Seite auf dem Bildschirm angezeigt werden soll.

Es gibt mehrere Möglichkeiten URIs zu verwalten. Eine zentrale Verwaltung wird zum Beispiel bei den URLs durch die DNS angewendet. Andere URIs je-

doch sind dezentral organisiert.

Aus dieser dezentralen Verwaltung entstehen jedoch wieder Probleme die in Kauf genommen werden müssen wenn man eine Technologie wie das Semantic Web generieren will. Zum Beispiel ist es bei URIs nicht möglich herauszufinden ob mehrere URIs die gleiche Resource bezeichnen. Das geschieht dadurch, dass jeder in der Lage ist ohne besondere Erlaubnis URIs zu erstellen, sogar für Ressourcen die ihm nicht selber gehören.

Im Grunde genommen ist ein URI nichts weiter als ein Bezeichner für eine Resource. Es wird keine Aussage darüber getroffen ob dieser Bezeichner bzw. die Ressource über das Web zugänglich ist oder nicht und es wird ebenfalls keine Aussage darüber getroffen ob über den Bezeichner mehr über die Ressource herausgefunden werden kann.

Das Semantic Web sieht vor, dass jede Ressource über eine URI verfügt.

**RDF** RDF bildet, wie auch im Abschnitt über die Architektur des Semantic Web schon angesprochen, eine Möglichkeit Metadaten in maschinenlesbarer Form darzustellen und auszutauschen. Des Weiteren bietet es einen Mechanismus um Ressourcen domänenneutral zu beschreiben.

RDF ist sehr vielseitig und kann daher in den unterschiedlichsten Bereichen zum Einsatz gebracht werden. Zum Beispiel kann es bei Suchmaschinen eingesetzt werden um deren Treffergenauigkeit zu erhöhen. Es kann zum Extrahieren des Inhalts und der inhaltlichen Verbindungen einer Webseite oder einer digitalen Bücherei angewendet werden.[6]

Bevor nun im Detail die Funktionsweise von RDF beschrieben wird, werden zuerst die Nachteile angesprochen, die XML gegenüber RDF hat. Dadurch wird klar, warum XML alleine nicht für das Semantic Web genügt.

Ein Vorteil von RDF besteht darin, dass sich die Autoren von Metadaten nur auf ein Schema beziehen müssen, welches erweiterbar und gemeinsam nutzbar ist.

RDF ist objektorientiert und die Schemata sind Klassen in dieser Umgebung. Durch den hierarchischen Aufbau ist es möglich ein Schema welches sich nur geringfügig von einem bestehenden unterscheidet durch inkrementelle Veränderungen zu erstellen. Das ist bei DTDs in einer XML Umgebung nicht möglich. Durch diese inkrementelle Veränderung der Schemata und deren Möglichkeit RDF Instanzen aus Schemata verschiedener Quellen zu bilden wird es möglich, dass Agenten sowohl auf den Instanzen an sich arbeiten können als auch auf dem Kombination die mächtiger ist als die Summe der Einzelteile.[7][8]

Ein weiteres Problem welches bei XML besteht ist, dass es keine eindeutige Art und Weise gibt wie XML Elemente verschachtelt werden. So gibt es für folgende Aussage mehrere Möglichkeiten der Darstellung in XML:

Der Student mit dem Namen "Max Mustermann" hat die Matrikelnummer "123456789".

Darstellungsart 1:

```
<student name="Max Mustermann">
  <matrikelnummer>123456789</matrikelnummer>
</student>
```

Darstellungsart 2:

```
<student>
  <matrikelnummer>123456789</matrikelnummer>
  <name>Max Mustermann</name>
</student>
```

Der Nachteil von XML liegt nun darin, dass XML zwar die Struktur eines Dokumentes vorgibt, jedoch nichts über die semantische Bedeutung des Inhalts eines Dokumentes weiß. Daher ist es schon in diesem einfachen Beispiel nicht möglich eine Eindeutige Darstellung zu finden.

RDF hingegen hat einen Satzbau der aus Tripeln mit *Subjekt-Prädikat-Objekt* besteht.

Um den oben dargestellten Fall als RDF Statement zu formulieren wird nun zuerst das Konzept einer Ressource erläutert. Wie im vorhergehenden Abschnitt schon erwähnt, kann eine Ressource alles sein, was durch eine URI eindeutig bestimmt werden kann.

*Subjekt:* Das Subjekt ist die Ressource über die eine Aussage gemacht werden soll. In diesem Beispiel handelt es sich um einen Studenten der über folgende URI identifiziert werden könnte: [http://www.ifi.lmu.de/mustermann\\_max](http://www.ifi.lmu.de/mustermann_max).

*Prädikat:* Das Prädikat definiert die Information die über das Subjekt gegeben werden soll. Im Beispiel wird über das Subjekt "Max Mustermann" die Aussage getroffen, dass er die Matrikelnummer "123456789" hat. Die Art der Information die über das Subjekt getroffen wird, wird wieder durch eine URI spezifiziert, die sich auch eine Ontologie bezieht <http://ifi.lmu.de/ontology#hatMatrNummer>. Was eine Ontologie ist und wie diese spezifiziert werden, wird später behandelt. Das Prädikat wird auch als Eigenschaft des Subjektes bezeichnet.

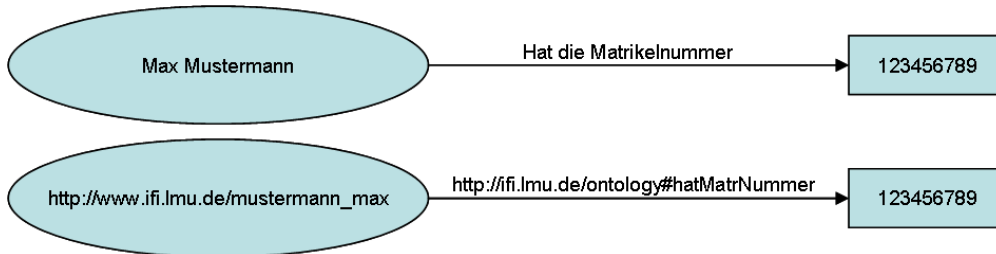
*Objekt:* Das Objekt bezeichnet den Wert des Prädikats. Im Beispiel ist der Wert der Matrikelnummer "123456789". Das Objekt kann entweder ein Literal sein oder eine andere Ressource die durch eine URI spezifiziert ist.

Anders betrachtet kann man das RDF Daten Modell auch als gerichteten Graph sehen. Dabei gilt folgende Notation:

- der eine Knoten bildet das Subjekt ab
- der andere Knoten bildet das Objekt ab
- die Kante vom Subjektknoten zum Objektknoten bildet das Prädikat

In dieser Darstellung würde das Beispiel aussehen wie in Abbildung 2. Für die reine RDF Darstellung gibt es unterschiedliche Möglichkeiten der Syntax. Für dieses Beispiel wäre folgender Ausdruck in RDF denkbar:

```
<http://www.ifi.lmu.de/mustermann\_max>
<http://ifi.lmu.de/ontology\#hatMatrNummer> "123456789".
```



**Abbildung 2.** RDF Graph für das Beispiel; Abbildung nach [9]

Jedes Statement wird als Folge von Subjekt, Prädikat und Objekt geschrieben und mit einem Punkt beendet.[9]

Um RDF jedoch standardisiert speichern und vor allem austauschen zu können gibt es auch eine Möglichkeit die Ausdrücke zu serialisieren und in eine Form von XML zu bringen. Das gegebene Beispiel würde in der angesprochenen RDF/XML Syntax folgende Form haben:

```
<?xml version="1.0">
<rdf:RDF
  xmlns:ifins"http://ifi.lmu.de/ontology#"
  xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description rdf:about="http://www.ifi.lmu.de/mustermann_max">
    <ifins:hatMatrNummer>123456789</ifins:hatMatrNummer>
  </rdf:Description>
</rdf:RDF>
```

Wie in diesem Beispiel deutlich wird ist die Kombination mehrerer Namespaces möglich. Entweder es werden selbst definierte Namespaces verwendet oder man greift auf offiziell standardisierte Namespaces wie das Dublin Core Schema zurück. Das folgende Beispiel zeigt zum einen wie Dublin Core eingebunden werden kann und zum anderen wie man komplexere Objekte mit mehreren Eigenschaften darstellen kann. Es wird wieder mit RDF/XML als Syntax gearbeitet:[10]

```
<?xml:namespace ns="http://www.w3.org/TR/WD-rdf-syntax"
  prefix="RDF"?>
<?xml:namespace ns="http://www.pur.org/metadata/dublin_core"
  prefix="DC"?>
<?xml:namespace ns="http://www.some.org/schemata/people"
  prefix="P"?>

<RDF:RDF
  <RDF:Description about="http://www.some.org/smith">
    <DC:Creator>
      <RDF:Description>
```



```

    <P:Name>John Smith</P:Name>
    <P:EMail>mailto:smith@some.org</P:EMail>
  </RDF:Description>
</DC:Creator>
<RDF:Description>
</RDF:RDF>

```

Der hieraus entstehende Graph ist in Abbildung 3 dargestellt.

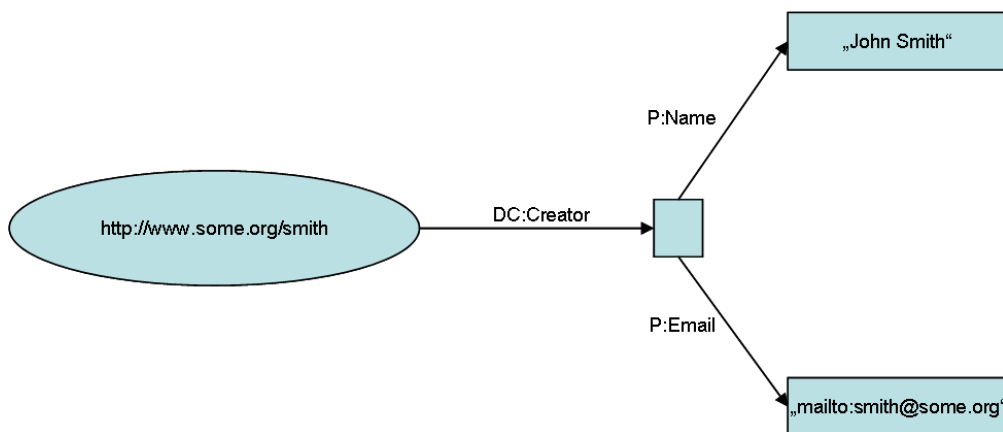


Abbildung 3. RDF Graph für das Beispiel; Abbildung nach [9]

**Ontologien: RDF Schema und OWL** Frühe Anwendungen von Ontologien sind vor allem im Bereich der künstlichen Intelligenz zu finden. Dort beschäftigen sich Wissenschaftler seit Anfang der 90er Jahre mit diesem Thema. Es gibt weit reichende Definitionen für diese Begrifflichkeit. Zusammenfassend kann man aber sagen, dass es sich bei Ontologien um die Beschreibung der Konzepte und Relationen in einer Domäne handelt. Sie bilden sozusagen das Vokabular das sowohl von Menschen als auch von Maschinen verstanden werden sollte. Auf einer abstrakten Ebene wird für alle zugreifenden Teilnehmer definiert was für die betreffende Domäne gilt. Ontologien bilden somit eine weitere wichtige Säule für das Semantic Web. Bevor nun beispielhaft die zwei wichtigsten Ontologien für das Semantic Web erläutert werden gibt es noch einige grundsätzliche Dinge zu Ontologien zu wissen. Im Allgemeinen bestehen Ontologien aus den folgenden drei Bausteinen:

1. Klassen einer Domäne
2. Beziehungen zwischen Klassen. Das können hierarchische Unterklassen, vordefinierte Beziehungen oder nutzerdefinierte Eigenschaften sein.
3. Einschränkungen, was ausgedrückt werden kann [11]

Des Weiteren müssen Ontologien folgende Voraussetzungen erfüllen um es möglich zu machen Domänen Modelle korrekt zu beschreiben:

1. Wohl definierte Syntax. Das ist wie bei allen Programmiersprachen eine wichtige Voraussetzung um eine Bearbeitung durch Maschinen möglich zu machen.
2. Wohl definierte Semantik. Eine ein-eindeutige Bedeutung der Ausdrücke ist unerlässlich. Es darf nicht sein, dass unterschiedliche Maschinen oder auch Menschen, die Semantik unterschiedlich deuten.
3. Effiziente Beweisführung. Es muss möglich sein nachzuweisen, dass eine Klasse zum Beispiel Äquivalent zu einer anderen ist, oder zu welcher Oberklasse eine gegebene Unterklasse gehört.
4. Ausreichende Ausdrucksstärke. Alle entscheidenden Ausdrücke für die entsprechenden Domäne müssen abbildbar sein.
5. Einfachheit der Ausdrücke. Die Ausdrücke müssen einfach zu verstehen sein, damit sie effizient eingesetzt werden können. [11]

Nun sind die Grundbausteine von Ontologien bekannt. Im Folgenden werden die beiden wichtigsten Ontologien für das Semantic Web erläutert. Dabei handelt es sich um RDF Schema (RDFS) und die Web Ontology Language (OWL).[11]

*RDF Schema:* RDFS ist eine sehr einfache Ontologie Sprache die dem Nutzer aber auch nicht viel Freiraum lässt. Es ist möglich über RDFS ein Vokabular einzurichten worüber es möglich wird die Ressourcen einer Domäne über Metadaten zu beschreiben. Wie auch bei RDF selber wird zu Beschreibung des RDF Schemas die Tripelschreibweise verwendet. Dem Nutzer werden die drei Bausteine Klassen, Eigenschaften und Beziehungen bereitgestellt um das Domänenmodell zu erstellen.

**Klassen:** Ressourcen werden unterteilt in so genannte Klassen. Die Instanzen einer Klasse werden als "set of instances" bezeichnet. Es ist möglich dass unterschiedliche Klassen dasselbe "set of instances" haben. Diese Klassen unterscheiden sich dann jedoch durch die Eigenschaften die für jede Klasse individuell angelegt werden.

**Eigenschaften:** Die Eigenschaften dienen dazu Beziehungen in der Ontologie darzustellen. Eigenschaften sind zum Beispiel "rdfs:subClassOf". Diese Eigenschaft beschreibt die Beziehung zwischen Ober- und Unterklasse. Eine Klasse kann Oberklasse mehrerer Unterklassen sein.

Eine komplette Liste der Klassen und Eigenschaften kann unter [12] gefunden werden.

Ontologien können wie auch RDF als Graphen abgebildet werden. Dadurch werden die herrschenden Beziehungen noch besser verdeutlicht. In Abbildung 4 ist eine mögliche Ontologie für das oben angeführte Studentenbeispiel abgebildet, der *ifilmu* Namespace sei vorausgesetzt. Es werden die Klassen *Mensch* und *Student* verwendet. Sie gehören zu der *rdfs:Class* Ressource und haben die *rdfs:Type* Eigenschaft. Die Klasse *Student* ist definiert als *rdfs:SubClassOf* der Klasse *Mensch*.

Außerdem werden in der Ontologie die Eigenschaften *hatMatrNr* und *istInSem* definiert.

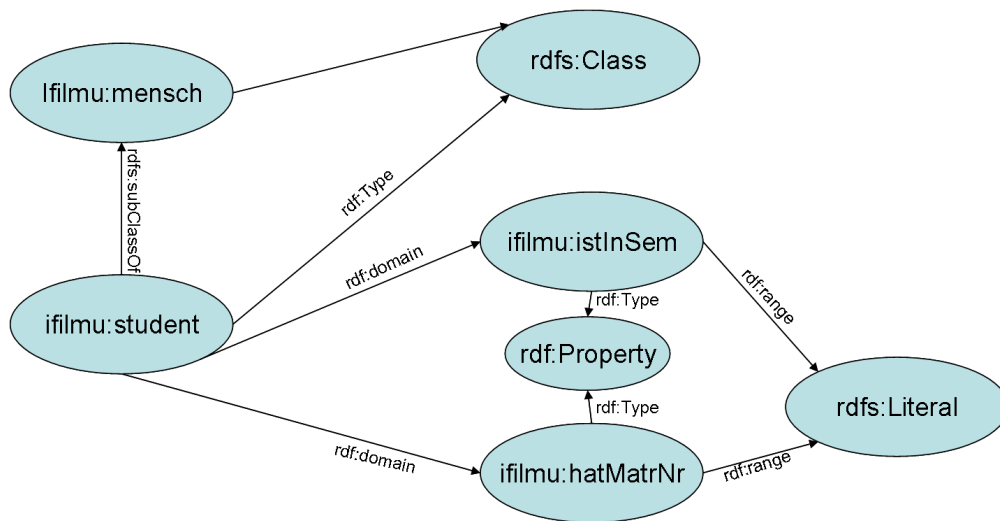


Abbildung 4. RDF Graph für die Beispielsontologie; Abbildung nach [9]

*OWL*: Auf Grund der Eingeschränktheit von RDF hat die Web Ontology Group des W3C eine Weiterentwicklung vorgenommen um eine mächtigere Ontologie Sprache zu schaffen. Als Ergebnis entstand DAML+OIL.[13] Aus dieser Sprache entwickelte sich dann die Web Ontology Language OWL, die nun vom W3C empfohlen wird. OWL besitzt eine Vielzahl von zusätzlichen Ausdrucksmöglichkeiten verglichen mit RDFS.

Um eine möglichst mächtige Sprache zu entwickeln die zum einen ausdrucksstark ist und zum Anderen ein hohes Maß an Integritätsunterstützung bietet wurde OWL in drei Untereinheiten aufgeteilt:

- OWL Lite: Bietet hierarchische Klassifizierung und einfache Beschränkungen, bei gleichzeitig geringer formalen Komplexität. Im Gegensatz zu RDFS kann man mit OWL Lite Gleichheiten aufstellen: *SameIndividual(Objekt1 Objekt2)*. Außerdem können Gleichheiten von Klassen direkt dargestellt werden: *EquivalentClasses(Class1 Class2)*.
- OWL DL: Bietet ein Höchstmaß an Ausdrucksstärke bei gleichzeitiger Garantie der Ausführbarkeit. OWL DL beinhaltet alle OWL Sprachkonstrukte, die aber teilweise nur mit gewissen Restriktionen benutzt werden können, um Korrektheit zu garantieren. Im Unterschied zu OWL Lite sind in OWL DL einige Konstrukte hinzugefügt die es zum Beispiel möglich machen auszudrücken, dass zwei Klassen Disjunkt sind was eine viel stärkere Aussage ist als die Ungleichheit: *DisjointClasses(Class1 Class2)*.  
Es lassen sich auch boolesche Ausdrücke mit einbeziehen. So kann man beispielsweise Ausdrücken, dass eine Klasse *Class1* nicht durch eine Klasse *Class2* oder *Class3* zusammengefasst wird, sondern durch deren Vereinigung: *SubClassOf(Class1 unionOf(Class2 Class3))*.
- OWL Full: Bietet ein Höchstmaß an Ausdrucksstärke und syntaktischer Freiheit. Es besteht jedoch keine Garantie auf Ausführbarkeit. In OWL Lite und OWL DL ist es nicht möglich, dass etwas Klasse und Instanz gleichzeitig ist. Dadurch wird die Ausführbarkeit gewährleistet. In OWL Full ist dies er-

laubt. Dadurch wird es möglich unterschiedliche Ontologien zu kombinieren in denen genau das der Fall ist. Deshalb ist es nötig eine Instanz mit einer Klasse gleichsetzen zu können.

Zwischen RDFS, OWL Lite und OWL DL lässt sich folgender mathematischer Zusammenhang herstellen:

$$RDFS \subset OWL\text{Lite} \subset OWL\text{DL}. [11]$$

Weitere Ontologie Sprachen sind zum Beispiel XOL, welche auf XML basiert, SHOE, welche auf HTML aufsetzt oder OML welche eine XML Serialisierung von SHOE darstellt. [9] [11] [14]

**Anwendungen des Semantic Web** In den vorangegangenen Kapiteln wurde beschrieben wie das Semantic Web aufgebaut ist, welche Technologien und Standards verwendet werden und welche Vorteile dadurch erzielt werden können.

Das alles hilft jedoch nichts, wenn es keine Anwendungen gibt, die es dem Nutzer auf der einen Seite möglich machen möglichst einfach und effizient Dokumente in den vorgegebenen Standards und Techniken zu entwickeln und auf der anderen Seite auf Dokumenten in diesen Formaten zu arbeiten.

Deshalb werden nun noch beispielhaft einige Technologien vorgestellt, die diese beiden Prozesse unterstützen.

Von der Firma Hewlett Packard und Brian McBride wurde ein Toolkit entwickelt welches auf Java basiert und dazu dient RDF Modelle zu erstellen, parsen und zu durchsuchen. Jena ist in der Lage RDF als XML zu speichern und aus einer XML Datei das entsprechende RDF Modell auszulesen. Jena wird in der Semantic Web Community als eines der führenden Werkzeuge zur Erstellung von Semantic Web Anwendungen betrachtet. HP will mit der Entwicklung dieses und anderer Werkzeuge wie Joseki, einer Web API für Jena [15], sowie BrownSauce, einem DRF Browser die Entwicklung des Semantic Web vorantreiben. Durch die Offenlegung der Sourcecodes sollen sich möglichst viele Entwickler beteiligen. [16] Das folgende Beispiel zeigt wie man mit Jena einen einfachen Graphen erstellen kann. Es wird dem Subjekt John Smith die Eigenschaft zugeordnet die seinen Namen widerspiegelt:

```
// Variablendefinitionen
static String personURI    = "http://somewhere/JohnSmith";
static String fullName     = "John Smith";

// Leeres Modell wird erzeugt
Model model = ModelFactory.createDefaultModel();

// Ressource wird erzeugt
Resource johnSmith = model.createResource(personURI);

// Eigenschaft wird zur Ressource hinzugefügt
johnSmith.addProperty(VCARD.FN, fullName);
```

Weitere Beispiele und die gesamte API von Jena kann unter [15] abgerufen werden.

Der strukturelle Aufbau des Semantic Web ist auch für andere Bereiche anwendbar. So gibt es zum Beispiel Ansätze, wie man die oben Beschriebenen Technologien auf MPEG-7 anwenden kann. Die University of Old in Australien hat hierfür ein Konzept entwickelt [17].

In der Forschung an diesem Institut wurde versucht mit RDF eine Ontologie für MPEG-7 zu entwickeln. Dazu wurden verschiedene Klassen gebildet die für multimediale Inhalte interessant sind. Die oberste Klasse ist der *Multimedia-Content*. Davon gibt es Unterklassen wie *Image*, *Video* oder *Audio*. Durch diese Struktur entsteht ein hierarchischer Baum der sich über RDF abbilden lässt. Des Weiteren sind Segmente definiert worden, die sich wiederum hierarchisch aufbauen.

Wie sich aber auch bei der Entwicklung des Semantic Web herausgestellt hat ist RDF nicht ausreichend um komplexe Strukturen abzubilden. Aus diesem Grund wurde auch in diesem Forschungsansatz die Ontologie für MPEG-7 auf DAML+OIL aufgebaut.

Sobald die Forschung auf diesem Gebiet abgeschlossen ist und ein MPEG-7 Ontologie definiert wurde ist es geplant den effektiven Austausch von Multimedia Daten über das Semantic Web möglich zu machen.

### 3 Zusammenfassung

In dieser Ausarbeitung werden zuerst die Umstände beschrieben in denen sich das Semantic Web entwickelt hat. Gleich darauf werden die verwendeten Begriffe motiviert und eine möglichst allgemein gültige Definition für Semantic Web eingeführt. Es wird darauf eingegangen aus welchen Technologien die heutzutage gängigen Standards entstanden sind. Danach wird in einem Überblick erklärt aus welchen Bestandteilen sich das Semantic Web aufbaut. Dabei werden die Technologien URI und RDF beschrieben. Es wird ebenfalls ein detaillierter Einblick in die Ontologie RDFS gewährt und erläutert warum diese Ontologie alleine nicht mächtig genug ist um das Semantic Web abbilden zu können. Nachdem die technologischen Aspekte geklärt sind wird beschrieben, welche Applikationen und Werkzeuge es heute schon gibt mit denen es möglich ist Anwendungen für Das Semantic Web zu gestalten. Dazu gehört auch ein Einblick, wie die Technologie des Semantic Web genutzt werden kann um die Entwicklung des MPEG-7 Standards voran zu trieben.

### Literatur

1. Holzinger, A.: Basiswissen Multimedia. Vogel (2000)
2. Dr. Wolfgang Dostal, Mario Jeckle, D.I.M.B.Z.: Semantic web. Technical report, ([www.objektspektrum.de](http://www.objektspektrum.de))
3. Liliana Cabral, John Domingue, E.M.T.P.F.H.: Approaches to Semantic Web Services: an Overview and Comparison. Volume 2004. (The Semantic Web: Research and Applications (ESWS 2004))
4. Vladimir Geroimenko, C.C.E.: Visualizing the Semantic Web. Springer (2003)
5. V. Richard Benjamins, Pmpeu Casanovas, J.B.A.G.E.: Law and the Semantic Web. Springer (2005)

6. W3C: RDF-Syntax: <http://www.w3.org/TR/PR-rdf-syntax/>. W3C (1999)
7. Henning, P.A.: Taschenbuch Multimedia. Fachbuchverlag Leipzig (2001)
8. P. Wittenburg, D.B.: Metadata overview and the semantic web. Technical report, (Max-Planck-Institute for Psycholinguistics Nijmegen)
9. Reif, D.I.G.: WEESA-Web Engineering for Semantic Web Applications. PhD thesis, Technische Universität Wien (2005)
10. Lassila, O.: Web metadata: A matter of semantics. Technical report, Nokia Research Center (1998)
11. Norbert Eisinger, J.M.E.: Reasoning Web. Springer (2005)
12. W3C: RDF-Schema: <http://www.w3.org/TR/rdf-schema/>. W3C (2004)
13. Dieter Fensel, Frank van Harmelen, I.H.D.L.M.P.F.P.S.: Oil: An ontology infrastructure for the semantic web. Technical report, (IEEE Intelligent Systems)
14. Stefan Decker, Frank van Harmelen, J.B.M.E.D.F.I.H.M.K.S.M.: The semantic web - on the respective roles of xml and rdf. Technical report, (Department of Computer Science, Stanford University, USA)
15. Research, H.L.S.W.: Jena Webseite: <http://jena.sourceforge.net/index.html>. (Sourceforge)
16. Research, H.L.S.W.: Semantic Web Labs von HP: <http://www.hpl.hp.com/semweb/index.htm>. (HP)
17. Hunter, J.: Adding multimedia to the semantic web - building an mpeg-7 ontology. Technical report, (University of Old, Australia)