

# Fachhochschule Wedel

Seminararbeit

in der Fachrichtung Medieninformatik

Wintersemester 2006/2007

## Die Beschreibungssprache RDF: Zweck, Aufbau, Beispiel

Bearbeitet von: Imken Chitralla  
J.-D.-Möller-Str. 86  
22880 Wedel  
Tel.: 04103 / 80 39 560  
[imken.chitralla@gmx.de](mailto:imken.chitralla@gmx.de)

Matr.-Nr.: MI2118

Erarbeitet im: 7. Verwaltungssemester

Abgegeben am: 13. Dezember 2006

Referent: Prof. Dr. Sebastian Iwanowski  
Fachhochschule Wedel  
Feldstr. 143  
22880 Wedel

# Inhaltsverzeichnis

Einleitung.....	3
RDF - Kurzbeschreibung.....	3
RDF-Modell und Syntax.....	4
Aufbau einer einfachen Aussage.....	4
Der RDF-Graph.....	5
Literele.....	6
Strukturierte Werte mittels „Blank Nodes“.....	6
Container in RDF.....	7
RDF-Repräsentationen.....	9
N-Triples.....	9
Notation 3 (N3).....	10
RDF/XML.....	11
Namensräume und Vokabulare in RDF/XML.....	12
Verwendung von RDF/XML im Web.....	13
RDF Schema.....	13
Was sind Klassen?.....	14
Was sind Properties?.....	16
RDF-Anwendungen.....	18
Dublin Core.....	18
RSS 1.0.....	19
RDFPic.....	21
Fazit.....	22
Abbildungsverzeichnis.....	23
Literaturverzeichnis.....	24

## Einleitung

Diese Seminararbeit beschäftigt sich mit dem vom WorldWideWeb Consortium (W3C) veröffentlichtem Standard „Resource Description Framework“ (RDF) zur Verarbeitung von Metadaten.

Neben dem RDF-Modell und der RDF-Syntax wird ferner noch auf Repräsentationsarten, insbesondere RDF/XML, der Erweiterung RDF Schema und einige Anwendungen eingegangen. Begrifflichkeiten, die zum Verständnis vom RDF hilfreich sind, werden erklärt werden. Grundkenntnisse in der Extensible Markup Language werden aber vorausgesetzt.

## RDF - Kurzbeschreibung

Das Resource Description Framework wird vom W3C mittlerweile durch sechs Dokumente definiert<sup>1</sup>:

- [RDF/XML Syntax Specification \(Revised\)](#), W3C Recommendation, Dave Beckett, ed.
- [RDF Vocabulary Description Language 1.0: RDF Schema](#), W3C Recommendation, Dan Brickley, R.V. Guha, eds.
- [RDF Primer](#), W3C Recommendation, Frank Manola, Eric Miller, eds.
- [Resource Description Framework \(RDF\): Concepts and Abstract Syntax](#), W3C Recommendation, Graham Klyne, Jeremy Carroll, eds.
- [RDF Semantics](#), W3C Recommendation, Patrick Hayes, ed.
- [RDF Test Cases](#), W3C Recommendation, Jan Grant, Dave Beckett, eds.

Es dient zusammen mit der Web Ontology Language als Grundstein des Semantic Web. Im August 1997 wurde der Entwurf des RDF vorgelegt und im Februar 1998 als Recommendation veröffentlicht. Nach einer Überarbeitung wurde die letztendliche Version 2004 vorgestellt und als fertig eingestuft.

Durch das RDF soll Computern ermöglicht werden, im Internet vorhandenes Wissen zu suchen und zu verarbeiten. Die geschieht, indem Webseiten

---

<sup>1</sup> Vgl. o.V: Resource Description Framework (RDF), Internet <http://www.w3.org/RDF/>, Stand: 27.10.2006, Abgerufen: 01.11.2006

(Ressourcen) mit Hilfe von Eigenschaften beschrieben werden.

## **RDF-Modell und Syntax**

Das reine RDF-Modell bietet in seiner Einfachheit nur die Möglichkeit, Beschreibungen über Ressourcen anzufertigen. Als Grundlage dafür dienen benannte Eigenschaften (named properties) und dazugehörige Eigenschaftswerte (property values). Eigenschaften sind sozusagen Attribute von Ressourcen, mit denen wir sogar Beziehungen zwischen ihnen darstellen können.

Die Darstellung erfolgt durch so genannte Aussagen (statements). Diese werden durch drei verschiedene Objekt-Arten aufgebaut:

1. Ressourcen (resources)
2. Eigenschaften (properties)
3. Eigenschaftswerte (values)

zu 1) Ressourcen repräsentieren das, was beschrieben werden soll. Sie werden über eine URI identifiziert, müssen aber nicht unbedingt für eine Webseite stehen.

zu 2) Bei den Eigenschaften handelt es sich um etwas, mit deren Hilfe Ressourcen charakterisiert werden können. Ihnen werden die im Punkt 3 genannten Werte zugewiesen, welche entweder Literale oder Verweise zu anderen Ressourcen sind.

### ***Aufbau einer einfachen Aussage***

Wie zuvor geklärt, bestehen Aussagen aus drei Teilen. Um den Sinn, der dahinter steht, besser zu verstehen, spricht man auch von Subjekt, Prädikat und Objekt. Dies lässt dann erkennen, dass es sich bei Aussagen um normale Sätze aus der natürlichen Sprache handelt. So aufgebaute Aussagen sind dann Ausgangsbasis für viele weitere Darstellungsarten.

**`http://www.fh-wedel.de` hat einen Titel mit Wert **Fachhochschule Wedel - Willkommen****

*Abbildung 1: Eine einfache Aussage mit Subjekt, Prädikat und Objekt (fett hervorgehoben)*

## **Der RDF-Graph**

Entwickelte Aussagen lassen sich am besten mit Hilfe eines gerichteten Graphen darstellen. Dabei werden die Statements durch Knoten und Pfeile repräsentiert. Hierbei gibt es nur einige wenige Dinge, die eingehalten werden müssen:

1) Knoten sind entweder

- eine URI
- ein Literal
- ein leerer Knoten (blank node)

2) Der Pfeil zeigt immer in Richtung des Objektes und stellt ein Prädikat dar.

Der RDF-Graph wird vom W3C als Standardentwicklungsmethode vorgeschlagen. Dies liegt nicht nur daran, dass dieser sehr schnell anzufertigen ist, sondern eher ausschlaggebend ist die sehr gute Lesbarkeit für den Menschen.

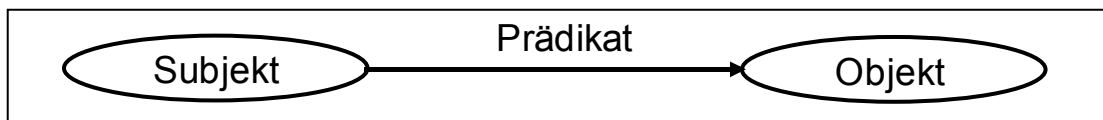


Abbildung 2: Aussage in einem RDF-Graph

Dies ist der syntaktische Aufbau einer RDF-Aussage, wenn dieser als Graph dargestellt wird. Schaut man sich dann nochmal die Aussage aus unserem ersten Beispiel in Abbildung 1 genauer an und entwickelt dazu den Graphen, fällt auf, dass dieser doch etwas von dem in Abbildung 2 eingeführten abweicht:

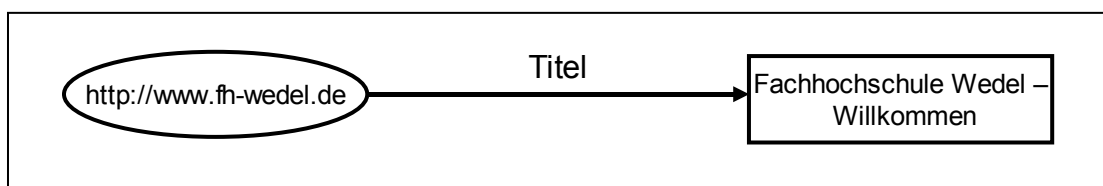


Abbildung 3: Beispiel Aussage als RDF-Graph

Das Objekt in diesem Beispiel wird als ein Literal dargestellt und nicht als eine URI. Dies ist im RDF-Modell möglich, wird aber in dem Graph dann als ein Rechteck dargestellt.

### ***Literale***

Doch was sind eigentlich Literale? Literale bieten die Möglichkeit, ein Objekt als ein in natürlicher Sprache verfassten Klartext darzustellen. Dieses bietet natürlich den Vorteil, dass sie um einiges intuitiver sind als URI. Nachteil ist dann aber, dass wir solche Objekte nicht durch weitere RDF-Aussagen beschreiben können und sie nicht eindeutig identifizierbar sind.

Literale können auf zwei verschiedene Arten angewendet werden. Es gibt „plain Literals“ und „typed Literals“. Der Unterschied zwischen den beiden ist aber recht eindeutig. Während ein „plain Literal“ nur ein einfacher String ist, mit deren Hilfe man etwas in natürlicher Sprache als Klartext darstellen kann, gibt man beim „typed Literal“ noch einen Datentyp mit an. Dies ist vor allem dann sinnvoll, wenn man mal überlegt, auf welche verschiedenen Arten der Wert „27“ interpretiert werden kann. Es ist ja nicht festgelegt, ob es sich dabei um eine Zahl oder aber um einen String mit den Zeichen „2“ und „7“ handelt. Hierbei sei aber zu beachten, dass eine Datentypspezifikation nicht Teil vom RDF ist. Um so etwas beim Verfassen von RDF-Dokumenten zu realisieren, muss auf XML Schema zurückgegriffen werden.

### ***Strukturierte Werte mittels „Blank Nodes“***

Möchten wir strukturierte Werte darstellen, ist das einfache RDF-Modell eigentlich etwas umständlich. Würden wir beispielsweise einer Person eine Adresse zuordnen, passierte das, indem wir aussagen würden, dass die Person eine Adresse hat, die Adresse eine Straße, eine Hausnummer, eine PLZ usw. hat. Das Problem hierbei ist die Ressource „Adresse“, denn für jede weitere Adresse müsste eine weitere URI angelegt werden. Allerdings benötigen wir diese globale Eindeutigkeit der „Adresse“, die eine URI mit sich bringt, gar nicht. Die eigentlich nur als eine Art Oberbegriff gedachte Ressource „Adresse“ kann weggelassen werden und durch einen leeren

Knoten ersetzt werden. Dabei wird erreicht, dass die Generierung einer neuen URI unnötig ist, aber trotzdem die Struktur der Werte erhalten bleibt. Weisen wir dann den „blank nodes“ noch einen Namen zu, können diese sogar innerhalb eines RDF/XML-Dokument eindeutig identifiziert werden.

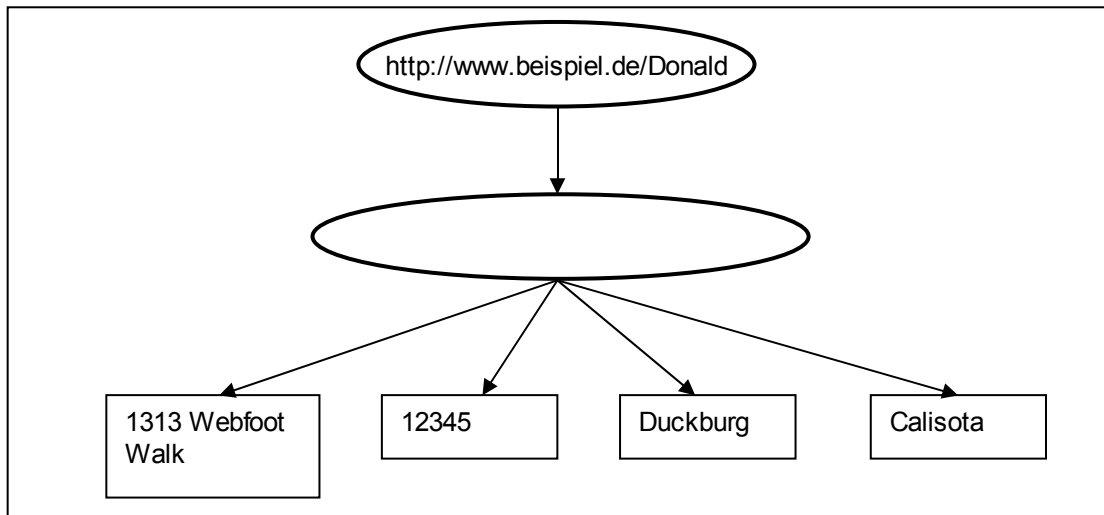


Abbildung 4: Darstellung von blank nodes

### **Container in RDF**

In RDF gibt es auch die Möglichkeit Gruppen, so genannte Container, zu beschreiben. Dafür gibt es das „Container Vocabulary“<sup>2</sup>, in dem drei verschiedene Arten definiert sind:

1. Bag
2. Sequence
3. Alternative

Bei der „Bag“ handelt es sich um eine Art Tasche. Hier können die Members, wie die Elemente eines Containers genannt werden, in ungeordneter Reihenfolge und auch mehrfach vorkommen.

Eine „Sequence“ unterscheidet sich nur in einem Punkt von der „Bag“: Hier ist die Reihenfolge der Werte signifikant.

<sup>2</sup> Vgl. Manola, Frank / Miller, Eric: RDF Primer, Internet <http://www.w3.org/TR/rdf-primer#containers>, Stand: 10.02.2004, Abgerufen: 27.11.2006

Die „Alternative“ bietet, wie der Name schon vermuten lässt, Alternativen an. Zur Verdeutlichung des Container-Prinzips soll das folgende Beispiel, übernommen aus dem RDF Primer<sup>3</sup>, helfen, indem eine mögliche Situation dargestellt wird, wofür die Container-Art „Bag“ eingesetzt werden könnte.

Das Beispiel in Abbildung 5 zeigt, wie die Hörschaft einer Vorlesung beschrieben werden kann. Mit Hilfe der Typ-Beschreibung wird die Eigenschaft „Studenten“ als eine „Bag“ beschrieben. Nun sind die folgenden Objekte Members des Containers.

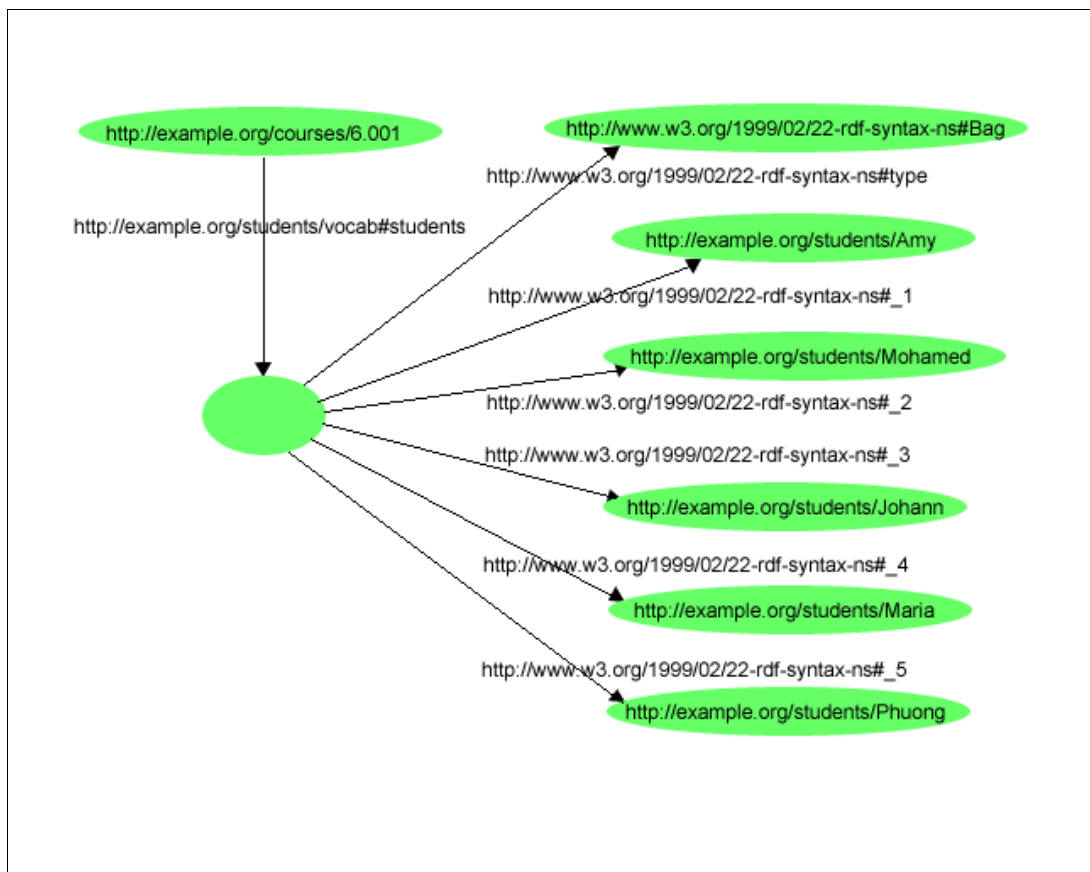


Abbildung 5: Darstellung einer Bag (vgl. Manola, Frank / Miller, Eric, RDF Primer, Internet <http://www.w3.org/TR/rdf-primer#containers>)

3 Vgl. Manola Frank / Miller, Eric: (FN 1)



## RDF-Repräsentationen

Als erstes sollte angemerkt werden: Das RDF-Modell ist unabhängig von der Syntax der Datenrepräsentation<sup>4</sup>.

Im folgenden werden dennoch drei weit verbreitete Darstellungsarten vorgestellt werden, wobei größtenteils aber nur die letzte neben der Darstellung auch zur Verarbeitung von RDF-Dokumenten genutzt wird. Sie alle bilden diesen Graph (Abbildung 6) ab.

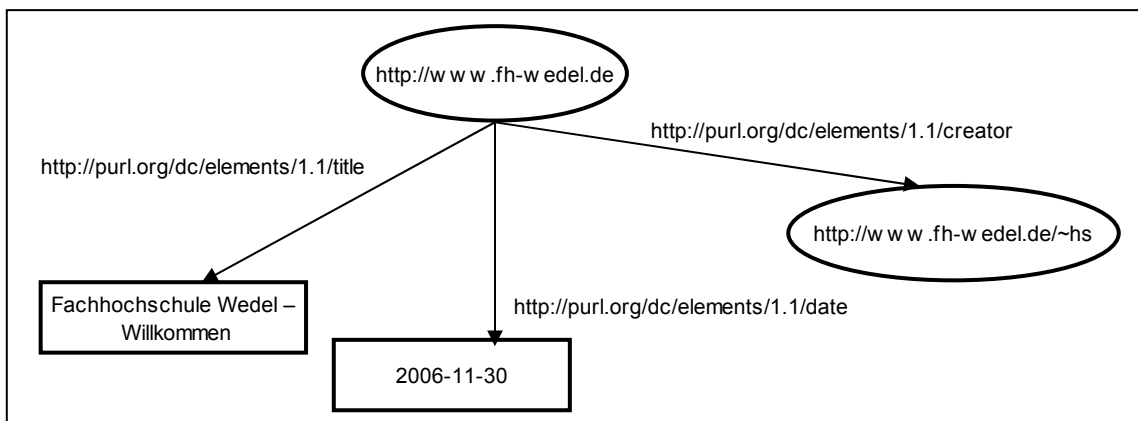


Abbildung 6: Umfangreichere Aussage als Beispiel für Repräsentationen

## N-Triples

Im Verlauf der Jahre hat sich N-Triples als besonders geeignet für das Erstellen und Prüfen von Testfällen<sup>5</sup> herausgestellt, da es in seiner Einfachheit überzeugt und dies vor allem für Maschinen von Vorteil ist.

Die Subjekt-Prädikat-Objekt-Tripel werden in dieser Repräsentationsart untereinander aufgelistet. Pro Zeile wird eine Aussage dargestellt. Mit einem . (Punkt) wird eine Aussage abgeschlossen.

Hier einmal unser Beispiel-Graph als N-Triples-Notation:

4 Böhme, Rainer: Problemseminar „Datenbankeinsatz im Internet“, Sommersemester 1999, Resource Description Framework, Internet <http://dbs.uni-leipzig.de/html/seminararbeiten/semSS99/arbeit5/Rdf.html>, Abschnitt:3.1, Abgerufen: 03.12.2006

5 Vgl. Ogbuji, Uche: Thinking XML: Introducing N-Triples, A simpler serialization for RDF, Internet <http://www-128.ibm.com/developerworks/xml/library/x-think17/index.html>, Stand: 08.04.2003, Abgerufen: 03.12.2006

```
<http://www.fh-wedel.de> <http://purl.org/dc/elements/1.1/title> "Fachhochschule Wedel - Willkommen" .  
<http://www.fh-wedel.de> <http://purl.org/dc/elements/1.1/date> "2006-11-30" .  
<http://www.fh-wedel.de> <http://purl.org/dc/elements/1.1/creator> <http://www.fh-wedel.de/~hs> .
```

### **Notation 3 (N3)**

Bei Notation 3 handelt es sich um eine recht kurze Syntax, die von Tim Berners-Lee entworfen wurde, dem „Vater“ des Internets. N3 ist eine sehr kompakte und gut lesbare Darstellungsform. Sie ist äquivalent<sup>6</sup> zu der im folgenden Abschnitt dargestellten RDF/XML-Syntax, aber vor allem für den Einstieg in RDF besser geeignet, da hier keine XML-Kenntnisse vorausgesetzt werden müssen.

Darüber hinaus ist die Darstellung mit einer der kürzesten, da Aussagen, in denen die gleichen Objekte oder Prädikate vorkommen, verkürzt dargestellt werden können.

Die Aussagen aus dem Abschnitt über N-Triples nun einmal in Notation3:

```
(1) @prefix dc: <http://purl.org/dc/elements/1.1/> .  
(2) @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
(3) <http://www.fh-wedel.de> dc:title "Fachhochschule Wedel - Willkommen" ;  
(4)    dc:date "2006-11-30" ;  
(5)    dc:creator <http://www.fh-wedel.de/~hs> .
```

Es ist sehr schön zu erkennen, wie wir die Aussage um einiges kürzer darstellen können. Durch die Verwendung von Namensräumen (Zeile 1 und 2), wird beim Aufruf der Prädikate (Zeile 3, 4, 5) auf die komplette URI verzichtet. Des Weiteren konnten die Prädikate „title“, „date“ und „creator“ nur durch Semikolon getrennt dem Subjekt „<http://www.fh-wedel.de>“ zugeordnet werden, so wurde zwar auch Platz eingespart, aber wichtigster Vorteil ist, dass die Struktur der RDF-Beschreibung in dieser Notation erkennbar ist.

<sup>6</sup> Vgl. Berners-Lee, Tim: Notation 3, Internet <http://www.w3.org/DesignIssues/Notation3>, Stand: 09.03.2006, Abgerufen: 03.12.2006

## **RDF/XML**

XML wurde zur Beschreibung von Bäumen entwickelt<sup>7</sup>. Da das RDF-Modell ein gerichteter Graph mit Subjekt-Prädikat-Objekt-Tripeln ist, bietet sich die Darstellung einer RDF-Beschreibung mit Hilfe von der Extensible Markup Language (XML) förmlich an. Die Verwendung von XML bietet sich auch aus dem Grund an, dass einfach vieles von der Sprache übernommen werden kann und dieses dann das Resource Description Framework gut ergänzt (siehe „Literale“).

Das zuvor verwendete Beispiel (Abbildung 6) als RDF/XML:

```
(1) <?xml version="1.0"?>
(2) <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
(3)         xmlns:dc="http://purl.org/dc/elements/1.1/">
(4)   <rdf:Description rdf:about="http://www.fh-wedel.de">
(5)     <dc:title>Fachhochschule Wedel - Willkommen</dc:title>
(6)     <dc:date>2006-11-30</dc:date>
(7)     <dc:creator rdf:resource="http://www.fh-wedel.de/~hs" />
(8)   </rdf:Description>
(9) </rdf:RDF>
```

Mit den Zeilen 2 und 9 geben wir an, dass alles, was sich dazwischen befindet, Beschreibungen in RDF sind. In den Zeilen 2 und 3 werden zwei Namensräume deklariert (ähnlich dem @prefix in N3). Dabei wird dem Präfix rdf und dem Präfix dc jeweils ein Schema zugewiesen.

Durch die Zeile 4 wird die eigentliche Beschreibung mit rdf:Description eingeleitet. Alles, was jetzt folgt, dient der schriftlichen Darstellung der Statements zur Beschreibung der Ressource „<http://www.fh-wedel.de>“.

Bei der RDF/XML-Repräsentation handelt es sich schon um eine verkürzte Darstellungsform. Voll ausgeschrieben würde das Dokument dann so aussehen:

---

<sup>7</sup> Walsh, Norman: Is RDF/XML Good for Anything?, Internet <http://norman.walsh.name/2004/07/30/rdfxml>, Stand: 11.09.2005, Abgerufen: 27.11.2006

```

( 1) <?xml version="1.0"?>
( 2) <rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
( 3)     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
( 4)     <rdf:Description rdf:about="http://www.fh-wedel.de">
( 5)         <dc:title>Fachhochschule Wedel - Willkommen</dc:title>
( 6)     </rdf:Description>

( 7)     <rdf:Description rdf:about="http://www.fh-wedel.de">
( 8)         <dc:date>2006-11-30</dc:date>
( 9)     </rdf:Description>

(10)     <rdf:Description rdf:about="http://www.fh-wedel.de">
(11)         <dc:creator rdf:resource="http://www.fh-wedel.de/~hs"/>
(12)     </rdf:Description>
(13) </rdf:RDF>

```

Jedem kann dabei aber selber überlassen werden, auf welche Art und Weise er sein RDF schreiben möchte, denn beide Beschreibungen sind vollständig äquivalent zueinander.

## Namensräume und Vokabulare in RDF/XML

Durch Namensräume werden in XML verschiedene Vokabulare beschrieben, bzw. unterscheidbar gemacht. In Vokabularen sind syntaktische Elemente enthalten, die helfen Strukturen zu bilden und die Bedeutung der Elemente und Attribute eingrenzen.

Begriffe können viele verschiedene Bedeutungen haben, je nachdem in welchem Kontext sie auftauchen. Dies führt schon unter den Menschen gelegentlich zu Kommunikationsproblemen, und bei Maschinen erst recht. Durch ein Vokabular (auch Schema) kann ausgedrückt werden, wie zum Beispiel „Titel“ aus unserem bisher verwendetem Beispiel interpretiert werden muss. Es wird also in diesem Fall gesagt, dass es sich um etwas handelt, was vergleichbar mit einem Buchtitel ist, jedoch aber nicht mit einem Titel für Personen, wie „Dr.“ oder „Frau“.

Mit der Hilfe von Vokabularen definieren wir demnach also die zur Beschreibung verfügbaren Eigenschaften, sagen, welche Bedeutung diese haben und können demnach die Semantik der Daten genauer beschreiben.

## Verwendung von RDF/XML im Web

Um RDF/XML in Dokumente einzubetten gibt es generell zwei verschiedenen Möglichkeiten. So kann eine RDF-Beschreibung extern gespeichert oder direkt in die Datei mit eingebunden werden.

Eine direkte Einbettung in einem HTML-Dokument sehe dann so aus:

```
(1) <head>
(2) <title>Irgendein Titel</title>
(3) <rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
(4)         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
(5)   <rdf:Description rdf:about="http://www.meinebeispielseite.de">
(6)     <dc:title>Meine Beispiel-Seite</dc:title>
(7)   </rdf:Description>
(8) </rdf:RDF>
(9) </head>
...
```

Wie man sehen kann, wird die RDF-Beschreibung einfach nur zwischen die `<head></head>`-Tags hineingeschrieben.

Wie aber auch bei der direkten Verwendung von CSS in HTML-Dokumenten, ist das auch hier nicht wirklich die eleganteste Lösung und die externe Variante mit XHTML ist der sicherste und sinnvollste Weg:

```
<link rel="alternate" type="application/rdf+xml" title="RDF Version"
href="RDFdatei.rdf" />
```

## RDF Schema

Das RDF-Modell beschreibt eigentlich nur Ressourcen, sagt aber nichts Näheres über die Charakteristik der Eigenschaften aus. Dies kann mit einem Schema (Vokabular) gemacht werden. Hier können wir für jede Eigenschaft festlegen<sup>8</sup>:

1. welche Bedeutung sie hat

<sup>8</sup> Vgl. Böhme, Rainer: Problemseminar „Datenbankeinsatz im Internet“, Sommersemester 1999, Resource Description Framework, Internet <http://dbs.uni-leipzig.de/html/seminararbeiten/semSS99/arbeit5/Rdf.html>, Abschnitt:3.2, Abgerufen: 03.12.2006

2. welche Werte erlaubt sind
3. welche Ressource-Arten diese Eigenschaft besitzen können
4. wie die Beziehung zu anderen Eigenschaften ist

Dies geschieht, indem wir einzelne Klassen definieren. Das Prinzip ist ähnlich dem Klassensystem objektorientierter Sprachen<sup>9</sup>: Ressourcen können Instanzen von einer oder auch mehreren Klassen sein und es lassen sich Hierarchien bilden. Anders aber als in objektorientierten Programmiersprachen, in denen es eine Art Korsett gibt, in das die Informationen passen müssen, bietet RDF Schema einfach nur die Möglichkeit, zusätzliche Informationen über die Ressourcen, die beschrieben werden, anzulegen<sup>10</sup>. Doch auch hier gibt es einen Nachteil (oder auch nicht): Es ist kein Mechanismus vorhanden, der dafür sorgt, dass alles ordnungsgemäß durchgeführt wird. Dies liegt also in den Händen des Verfassers.

Was bringt uns nun RDF Schema konkret: Bei RDF Schema handelt es sich um eine Art „Schema Definition Language“, mit deren Hilfe dann das eigentlich Schema (Vokabular) definiert wird. Hier wird dann festgelegt, was ein „Titel“ ist und welche Bedeutung diese Prädikat hat und welches Subjekt dies eigentlich haben darf (siehe „Namensräume und Vokabulare in RDF“).

### ***Was sind Klassen?***

Klassen stehen für Dinge, die beschrieben werden sollen. Sie können nahezu alles repräsentieren. Beschrieben werden sie durch `rdfs:Class`, `rdfs:Resource`, `rdf:type` und `rdfs:subClassOf`<sup>11</sup>.

Mit Hilfe von `rdf:type` und `rdfs:Class` lassen sich Instanzen aus dem abstraktem Objekt erzeugen. `rdfs:subClassOf` dient der Realisierung von Klassenhierarchien.

Das folgende Beispiel (Abbildung 7), übernommen aus dem RDF Primer<sup>12</sup>,

---

9 Vgl. Manola, Frank / Miller, Eric: RDF Primer, Internet <http://www.w3.org/TR/rdf-primer/#rdfschema>, Stand: 10.02.2004, Abgerufen: 27.11.2006

10 Vgl. Manola, Frank / Miller, Eric: (FN 8)

11 Vgl. Manola, Frank / Miller, Eric: RDF Primer, Internet <http://www.w3.org/TR/rdf-primer/#schemaclasses>, Stand: 10.02.2004, Abgerufen: 03.12.2006

12 Vgl. Manola, Frank / Miller, Eric: (FN10)

soll das Klassenprinzip einmal verdeutlichen. Es werden die Klassen „MotorVehicle“, „PassengerVehicle“, „Van“, „Truck“, „MiniVan“ erzeugt. Danach wird durch Gebrauch von `rdfs:subClassOf` eine Klassenhierarchie aufgebaut, die verdeutlicht, wie die Fahrzeug-Klassen strukturiert sind.

<code>ex:MotorVehicle</code>	<code>rdf:type</code>	<code>rdfs:Class .</code>
<code>ex:PassengerVehicle</code>	<code>rdf:type</code>	<code>rdfs:Class .</code>
<code>ex:Van</code>	<code>rdf:type</code>	<code>rdfs:Class .</code>
<code>ex:Truck</code>	<code>rdf:type</code>	<code>rdfs:Class .</code>
<code>ex:MiniVan</code>	<code>rdf:type</code>	<code>rdfs:Class .</code>
<code>ex:PassengerVehicle</code>	<code>rdfs:subClassOf</code>	<code>ex:MotorVehicle .</code>
<code>ex:Van</code>	<code>rdfs:subClassOf</code>	<code>ex:MotorVehicle .</code>
<code>ex:Truck</code>	<code>rdfs:subClassOf</code>	<code>ex:MotorVehicle .</code>
<code>ex:MiniVan</code>	<code>rdfs:subClassOf</code>	<code>ex:Van .</code>
<code>ex:MiniVan</code>	<code>rdfs:subClassOf</code>	<code>ex:PassengerVehicle .</code>

*Abbildung 7: Beispiel einer Klassenstruktur (Vgl. Manola, Frank / Miller, Eric: RDF Primer, Internet <http://www.w3.org/TR/rdf-primer/#schemaclasses>, Stand: 10.02.2004, Abgerufen: 03.12.2006)*

In RDF/XML wird das dann so realisiert: (siehe nächste Seite)

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://example.org/schemas/vehicles">

  <rdf:Description rdf:ID="MotorVehicle">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>

  <rdf:Description rdf:ID="PassengerVehicle">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdf:Description>

  <rdf:Description rdf:ID="Truck">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdf:Description>

  <rdf:Description rdf:ID="Van">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdf:Description>

  <rdf:Description rdf:ID="MiniVan">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Van"/>
    <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
  </rdf:Description>
</rdf:RDF>

```

## ***Was sind Properties?***

Nur mit Klassen lassen sich keine Ressourcen beschreiben. Dafür werden dann die Eigenschaften benötigt.

Properties werden dazu verwendet, Klassen zu charakterisieren und zu definieren, welche Eigenschaften für die zu beschreibenden Dinge verwendet werden sollten. Dafür werden `rdf:Property`, `rdfs:domain`, `rdfs:range` und `rdfs:subPropertyOf` eingesetzt<sup>13</sup>.

---

<sup>13</sup> Vgl. Manola, Frank / Miller, Eric: RDF Primer, Internet <http://www.w3.org/TR/rdf-primer/#properties>, Stand: 10.02.2004, Abgerufen: 03.12.2006



Bei `rdf:Property` handelt es sich um die Basisklasse für Eigenschaften. Durch `rdfs:subPropertyOf` lassen sich, wie auch bei `rdfs:subClassOf`, Vererbungshierarchien festlegen. `rdfs:range` und `rdfs:domain` sind dazu da, Werte einzuordnen. Genauer gesagt: mit `rdfs:range` wird gesagt, dass die Werte Instanzen einer bestimmten Klasse sind und mit `rdfs:domain` können wir den Anwendungsbereich einer Eigenschaft festlegen, was auch sehr sinnvoll ist, wenn man mal bedenkt, dass in RDF Schema die Eigenschaften unabhängig von Klassen definiert werden. Wenn wir also die Eigenschaft „Gewicht“ haben, diese mehrere Anwendungsbereiche hat (Klasse „Buch“ und Klasse „Fahrzeuge“), dann würde automatisch jede Ressource, die die Eigenschaft „Gewicht“ besitzt, ein Buch UND ein Fahrzeug sein<sup>14</sup>. Hier ist also Vorsicht geboten.

---

<sup>14</sup> Vgl. Manola, Frank / Miller, Eric: (FN12)

## **RDF-Anwendungen**

RDF ist Grundlage für die Verarbeitung von Metadaten in diversen Anwendungsbereichen. Dazu zählt:

1. Die Beschreibung des Inhaltes von Webseiten, Verzeichnissen, Datenbanken
2. Verbesserung der Suchfunktionen
3. Content Rating
4. Angaben von Urheberrecht
5. E-Commerce => Käufer/ Verkäufer finden, Vertragsbedingungen austauschen...
6. Wissensaustausch mit intelligenten Software-Agenten

Im folgenden werden drei Anwendungen kurz vorgestellt: Dublin Core, RSS1.0 und RDFPic.

### ***Dublin Core***

Bei „Dublin Core“ handelt es sich um ein Format, welches anfangs zur Beschreibung von Büchern eingesetzt wurde<sup>15</sup>. Mitte der neunziger Jahre wurde in der Bibliotheks-Welt der Wunsch nach elektronischen Suchsystemen laut. Dazu sollte ein einheitliches Format zur Beschreibung der Metadaten entwickelt werden. Auf dem „Metadata Workshop“ in Dublin, Ohio, entstand dann das „Dublin Core Format“ und wird seit dem durch die „Dublin Core Metadata Initiative“ weiter betreut.

Der Grund, weswegen das Format so weit verbreitet ist (auch hier wurde in den Beispielen das „Dublin Core“-Format verwendet), ist wohl das sehr einfache Vokabular. Definiert werden insgesamt 15 Eigenschaften (Abbildung 8), mit denen vor allem die gängigsten Dinge im Web beschrieben werden können: nämlich Webseiten.

---

<sup>15</sup> Vgl. Hjelm, Johan, 2001: Creating the Semantic Web with RDF, S. 4f

title	publisher	format	relation
creator	contributer	identifier	coverage
subject	date	source	rights
description	type	language	

Abbildung 8: Eigenschaften des Dublin Core Formats

Es folgt ein Beispiel einer RDF/XML Beschreibung, in der ausschließlich das „Dublin Core“-Vokabular verwendet wird:

```
( 1) <rdf:RDF
( 2)   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
( 3)   xmlns:dc="http://purl.org/dc/elements/1.1/">
( 4)   <rdf:Description rdf:about="http://www.seminarbeispiel.de">
( 5)     <dc:title>Mein kleines DC-Seminar-Beispiel</dc:title>
( 6)     <dc:description>Die Seite dient ausschließlich dazu, ein Beispiel für
( 7)       das Dublin Core Format zu zeigen</dc:description>
( 8)     <dc:publisher>Imken Chitralla</dc:publisher>
( 9)     <dc:date>2006-12-02</dc:date>
(10)     <dc:type>World Wide Web Home Page</dc:type>
(11)     <dc:format>text/html</dc:format>
(12)     <dc:language>de</dc:language>
(13)   </rdf:Description>
(14) </rdf:RDF>
```

In Zeile 3 wird der Namensraum für das „Dublin Core“ definiert und in der folgenden Beschreibung können die Eigenschaft mit dem Präfix genutzt werden.

## **RSS 1.0**

RSS steht für RDF Site Summary und ist wohl die am besten entwickelte RDF-Applikation im Web<sup>16</sup>. Mit Hilfe von RDF werden Nachrichten über das Internet versendet und mit Newsreadern empfangen. Beschreibungen von Nachrichten werden gespeichert und in maschinenlesbarer Form bereitgestellt. Ohne hunderte von Webseiten besuchen zu müssen, können die für den Anwender interessanten Informationen verfolgt werden.

<sup>16</sup> Vgl. Manola, Frank / Miller, Eric: RDF Primer, Internet <http://www.w3.org/TR/rdf-primer/#rss>, Stand: 10.02.2004, Abgerufen: 03.12.2006



Abbildung 9: Darstellung eines RSS-Feeds in Thunderbird

Wie groß die Bedeutung von RSS 1.0 ist, wird dadurch deutlich, dass es sich dabei sogar um den Standardnamensraum in RDF handelt. Wird also keine Namensraumangabe gemacht, gehen die Anwendungen davon aus, dass es sich um RSS handelt!

channel	link	name
image	description	url
item	items	textInput

Abbildung 10: Eigenschaften des RDF Site Summary

Mittlerweile allerdings wurde das RSS 1.0 zum Großteil durch RSS 2.0

abgelöst. Hierbei handelt es sich nun leider nicht mehr um eine RDF-Beschreibung.

## **RDFPic**

Bei RDFPic wurde zu Demonstrationszwecken vom W3C entwickelt<sup>17</sup> und verwendet RDF zur Beschreibung von Fotos. Bilder können mit Metadaten versehen werden, so dass beispielsweise eine schnellere und komfortablere Suche in elektronischen Fotoalben möglich ist.

Eine Foto-Beschreibung setzt sich in RDFPic aus drei Teilen zusammen:

1. Beschreibungen mittels Dublin Core (title, description, creator, date, type)
2. Technische Beschreibungen (camera, film, lens, devel-date)
3. Inhaltliche Beschreibungen (Portrait, Group-Portrait, Landscape, Baby, Architecture, Wedding, Macro, Graphic, Panorama, Animal)

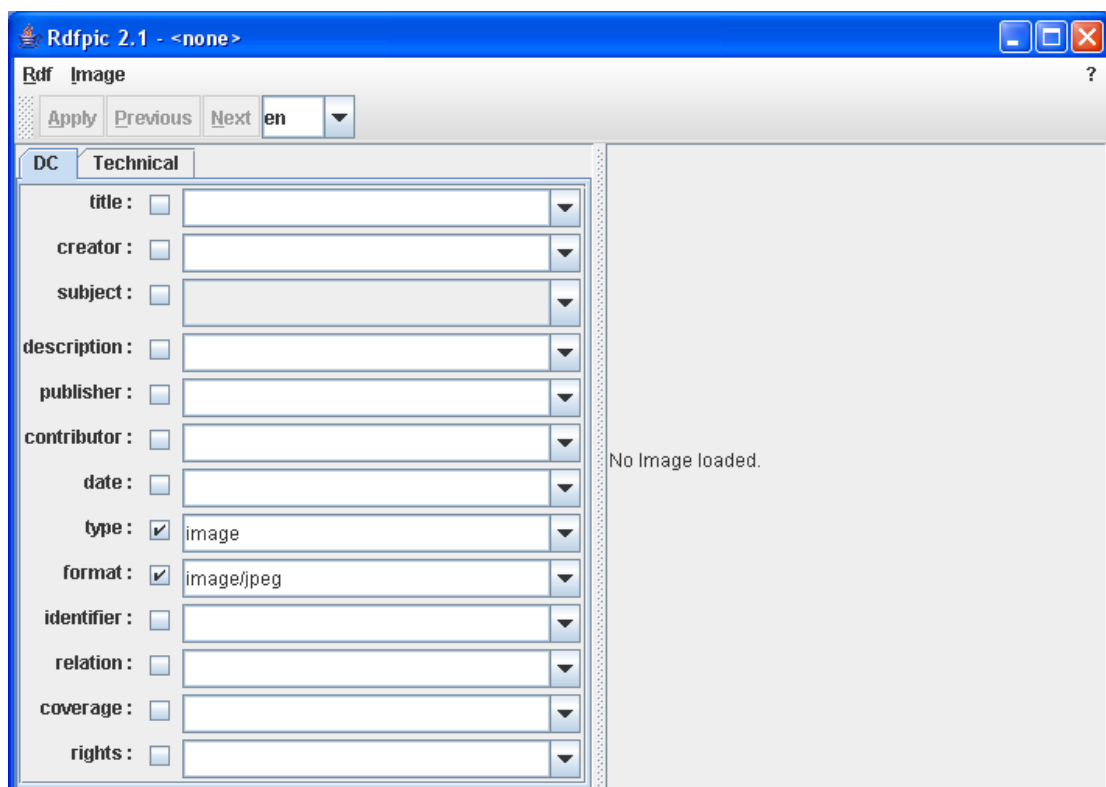


Abbildung 11: rdfpic 2.1 von Bert Bos entwickelt

<sup>17</sup> Vgl. Hjelm, Johan, 2001: Creating the Semantic Web with RDF, S. 23ff

## **Fazit**

Das Resource Description Framework ist eine gute Sache. Ressourcen im Web können nahezu ideal beschrieben werden, welches vor allem in Hinsicht auf das große Ziel „Semantic Web“ von großer Bedeutung ist.

In manchen Bereichen geht RDF aber nicht weit genug. Auch mit dem RDF Schema lässt sich nicht alles realisieren, weswegen auch noch weitere Ontologie-Sprachen entwickelt wurden.

Allerdings lässt sich RDF für das „Standard-Web“ gut verwenden, sofern es nur um generelle Beschreibungen geht, aber auch das wäre schon in der Unorganisiertheit des Internets ein riesiger Fortschritt. Was nützt es zu wissen, dass man eine Antwort im Netz finden kann, wenn man nicht an die geeigneten Informationen herangeführt wird. Hier könnte das Resource Description Framework eine gute Hilfe sein. Doch was bringt das Ganze, wenn es nicht von jedem Webseitenbetreiber angewendet wird? Und genau das ist die momentane Situation im Internet...

## Abbildungsverzeichnis

Abbildung 1: Eine einfache Aussage mit Subjekt, Prädikat und Objekt (fett hervorgehoben) .....	4
Abbildung 2: Aussage in einem RDF-Graph.....	5
Abbildung 3: Beispiel Aussage als RDF-Graph.....	5
Abbildung 4: Darstellung von blank nodes.....	7
Abbildung 5: Darstellung einer Bag (vgl. Manola, Frank / Miller, Eric, RDF Primer, Internet <a href="http://www.w3.org/TR/rdf-primer#containers">http://www.w3.org/TR/rdf-primer#containers</a> ).....	8
Abbildung 6: Umfangreichere Aussage als Beispiel für Repräsentationen.....	9
Abbildung 7: Beispiel einer Klassenstruktur (Vgl. Manola, Frank / Miller, Eric: RDF Primer, Internet <a href="http://www.w3.org/TR/rdf-primer/#schemaclasses">http://www.w3.org/TR/rdf-primer/#schemaclasses</a> , Stand: 10.02.2004, Abgerufen: 03.12.2006).....	15
Abbildung 8: Eigenschaften des Dublin Core Formats.....	19
Abbildung 9: Darstellung eines RSS-Feeds in Thunderbird.....	20
Abbildung 10: Eigenschaften des RDF Site Summary.....	20
Abbildung 11: rdfpic 2.1 von Bert Bos entwickelt.....	21

## Literaturverzeichnis

- o.V: Resource Description Framework (RDF),  
Internet <http://www.w3.org/RDF/>, Stand: 27.10.2006, Abgerufen:  
01.11.2006
- Berners-Lee, Tim: Notation 3, Internet  
<http://www.w3.org/DesignIssues/Notation3>, Stand: 09.03.2006,  
Abgerufen: 03.12.2006
- Böhme, Rainer: Problemseminar „Datenbankeinsatz im Internet“,  
Sommersemester 1999, Resource Description Framework, Internet  
[http://dbs.uni-  
leipzig.de/html/seminararbeiten/semSS99/arbeit5/Rdf.html](http://dbs.uni-leipzig.de/html/seminararbeiten/semSS99/arbeit5/Rdf.html),  
Abgerufen: 03.12.2006
- Hjelm, Johan, 2001: Creating the Semantic Web with RDF, WILEY
- Manola, Frank / Miller, Eric: RDF Primer,  
Internet <http://www.w3.org/TR/rdf-primer> Stand: 10.02.2004,  
Abgerufen: 27.11.2006
- Ogbuji, Uche: Thinking XML: Introducing N-Triples, A simpler  
serialization for RDF, Internet [http://www-  
128.ibm.com/developerworks/xml/library/x-think17/index.html](http://www-128.ibm.com/developerworks/xml/library/x-think17/index.html), Stand:  
08.04.2003, Abgerufen: 03.12.2006
- Walsh, Norman: Is RDF/XML Good for Anything?, Internet  
<http://norman.walsh.name/2004/07/30/rdfxml>, Stand: 11.09.2005,  
Abgerufen: 27.11.2006