

Het Semantische Web  
Technieken voor de toekomst van het WWW

Dennis Janse, 0213861, [djanse@sci.ru.nl](mailto:djanse@sci.ru.nl)

19 juni 2006

# Inhoudsopgave

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Inleiding</b>                             | <b>2</b>  |
| 1.1      | Probleemstelling en verantwoording . . . . . | 2         |
| 1.2      | Theoretisch kader . . . . .                  | 2         |
| 1.3      | Methode . . . . .                            | 2         |
| 1.3.1    | Onderzoeksfunctie . . . . .                  | 2         |
| 1.3.2    | Deelvragen . . . . .                         | 2         |
| <b>2</b> | <b>Op weg naar het WWW</b>                   | <b>4</b>  |
| 2.1      | De eerste ideeën . . . . .                   | 4         |
| 2.2      | Eerdere ontwikkelingen . . . . .             | 5         |
| 2.3      | Het uiteindelijke WWW . . . . .              | 5         |
| <b>3</b> | <b>Doel van het SW</b>                       | <b>6</b>  |
| 3.1      | Problemen . . . . .                          | 6         |
| 3.2      | Semantiek in het SW . . . . .                | 7         |
| <b>4</b> | <b>Technieken</b>                            | <b>9</b>  |
| 4.1      | XML . . . . .                                | 9         |
| 4.2      | XML Namespaces . . . . .                     | 9         |
| 4.3      | RDF . . . . .                                | 10        |
| 4.4      | RDF Schema . . . . .                         | 12        |
| <b>5</b> | <b>Conclusie</b>                             | <b>16</b> |

# Hoofdstuk 1

## Inleiding

### 1.1 Probleemstelling en verantwoording

Het web is nu zo'n 17 jaar oud. Het World Wide Web (WWW) is in 1989 uitgevonden<sup>1</sup>. Langzaam maar zeker, en steeds sneller groeide het Web tot wat het nu is. Het zal zeker ook nog verder groeien. Daardoor zal informatie moeilijker te vinden zijn. Hoe zorgen we ervoor dat deze enorme berg aan informatie toch betekenis voor ons heeft?

Ik wil in deze scriptie onderzoeken hoe het idee van het *Semantische Web* (SW) een rol speelt bij het vinden en uitwisselen (tussen mens en machine of machines onderling) van informatie in het WWW. Hoe is het ontstaan en wat wil men hiermee bereiken? Welke technieken spelen een rol, en kunnen die technieken dat doel ook waar maken? Voorlopig zal ik het Semantische Web definiëren als het web van informatie dat niet alleen voor mensen, maar ook voor machines begrijpelijk is.

Hieruit volgt dus dat ik een literatuurstudie wil doen naar het begrip SW. Eerst wil ik het begrip definiëren. Hiervoor is een uitstap nodig naar de filosofie achter het idee en daarmee ook naar de geschiedenis van het (Semantische) Web. Wat zijn de doelen van het SW? Vervolgens wil ik bekijken welke technieken een rol spelen en hoe deze aansluiten op de filosofie achter het SW. Als laatste wil ik antwoord geven op de vraag of het SW de gestelde doelen kan behalen.

### 1.2 Theoretisch kader

Uit de naam, het *Semantische Web*, volgt al dat deze scriptie te maken heeft met semantiek. Met name het aantonen dat de gebruikte technieken de doelen van het SW kunnen verwezenlijken, vereist een benadering vanuit het vakgebied van de semantiek. Daarnaast heeft de scriptie ook een grote filosofische inslag door in te gaan op het waarom van het SW.

### 1.3 Methode

#### 1.3.1 Onderzoeksfunctie

Deze scriptie heeft vooral een beschrijvende onderzoeksfunctie. Door de doelen van het SW helder te maken en daarna te kijken of de gebruikte technieken die doelen kunnen verwezenlijken, zal ik een antwoord kunnen geven op de vraag of het SW verwezenlijkt kan worden.

#### 1.3.2 Deelvragen

1. Wat is de geschiedenis van het SW?

---

<sup>1</sup>Let op, dit is iets anders dan Internet. Het is de "universal information space", opgebouwd uit documenten die aan elkaar gekoppeld zijn met hypertext. Dit alles is gebouwd óp het Internet.

2. Wanneer is dit begrip voor het eerst genoemd? Speelde het daarvóór al een rol?
3. Welke filosofie zit er achter het SW?
4. Welke doelen worden er gesteld door deze filosofie?
5. Zijn deze doelen bereikt?
6. Zullen deze doelen bereikt worden? Binnen welke termijn?
7. Welke technieken spelen een rol bij het SW?
8. Hoe zijn die technieken ontstaan?
9. Is er gebruik gemaakt van al bestaande technieken?
10. Wat voor functie hebben de technieken? Op welke manier kennen zij semantiek toe aan het web?

## Hoofdstuk 2

# Op weg naar het WWW

Wanneer we iets willen begrijpen van het Web en van zijn toekomstige mogelijkheden, zullen we moeten kijken hoe het tot stand kwam. In dit eerste deel van deze scriptie willen we daarom ingaan op het ontstaan van het World Wide Web (WWW). Omdat het ontstaan van het WWW zich voor het grootste deel afspeelt rond Tim Berners-Lee, zullen we hier dan ook veel gebruik maken van zijn boek: *“Weaving the Web”*. Doordat we beter zullen begrijpen wat er vooraf ging aan het ontstaan van het Web, zullen we daarna ook meer kunnen zeggen over het Semantische Web.

### 2.1 De eerste ideeën

De uitvinding van het Web was niet zozeer één bepaald punt in de tijd, maar meer een groeiend bewustzijn dat het schikken van informatie op een ongebonden “web-achtige” manier erg krachtig kan zijn. Het eerste deel van het verhaal speelt zich af rond Tim Berners-Lee, de uitvinder van het WWW. Zijn ouders waren allebei betrokken bij het team dat ’s werelds eerste commerciële stored-program computer ontwikkelde. Al in zijn jeugd was hij bezig met het idee dat computers een stuk krachtiger zouden zijn als ze data kunnen koppelen die in eerste instantie niet gekoppeld is. Computers houden namelijk al hun data in strakomlijnde matrices en hiërarchieën, alles is logisch geordend. Het menselijk brein daarentegen werkt veel meer met random associaties. Als ik bijvoorbeeld een muziknummer hoor op de radio dat tijdens mijn vorige vakantiebaan een hit was, dan denk ik gelijk terug aan die vakantiebaan.

In 1980 kreeg Tim Berners-Lee werk bij CERN. Hier schreef hij *Enquire*, zijn eerste “web-achtige” programma. Dit was een programma om informatie over personen, computers en projecten aan elkaar te kunnen koppelen. Er waren namelijk nogal wat verschillende soorten computers, besturingssystemen en talen in gebruik. De gedachte dat alle data in alle computers gekoppeld zou kunnen worden bleef hem bezig houden: *“Suppose I could program my computer to create a space in which anything could be linked to anything.”* [4, blz. 4] Elk stukje informatie zou dan een adres moeten krijgen, zodat een computer het op zou kunnen halen.

*“Computers might not find the solutions to our problems, but they would be able to do the bulk of the legwork required, assisting our human minds in intuitively finding ways through the maze. The added excitement was that computers also could follow and analyze the tentative connective relationships that defined much of our society’s workings, unveiling entirely new ways to see our world.”*[4, blz. 5]

We zien hier dat er al in het begin het idee van “machine-understandable” (machine begrijpelijke) data was. Computers zouden de koppelingen tussen verschillende documenten kunnen volgen en analyseren.

## 2.2 Eerdere ontwikkelingen

Er was al eerder nagedacht over koppelen van informatie, bijvoorbeeld door Vannevar Bush in 1945 [6]. Die schreef over een mechanische machine (de Memex) die cross-references tussen microfilms zou kunnen maken.

Daarnaast was er Ted Nelson, die in al in 1965 schreef over “Literary Machines” [10]. Computers die mensen in staat stellen om op een nieuwe manier teksten te schrijven, hypertext genaamd. Dit soort teksten hoef je niet in een bepaalde volgorde te lezen, de lezer kan door hyperlinks te volgen allerlei andere stukken tekst uit andere documenten raadplegen. Het idee was dat die links twee kanten op werken, zodat de auteur van een stuk tekst elke keer dat er naar zijn tekst gelinkt wordt een kleine vergoeding kan krijgen voor het quoten van zijn tekst.

Ook was er nog Doug Engelbart. Zijn visie was dat mensen hypertext zouden gebruiken als een hulpmiddel om samen te werken. Hiervoor vond hij de muis uit (in de jaren 60 van de 20<sup>e</sup> eeuw), zodat je makkelijk hyperlinks kon selecteren in een tekst.

De volgende belangrijke ontwikkeling was het ontstaan van het Internet, dat verschillende netwerken van computers aan elkaar verbond. Het Internet stamt uit de jaren 70 van de 20<sup>e</sup> eeuw.

## 2.3 Het uiteindelijke WWW

Een aantal punten zijn van cruciaal belang geweest bij de ontwikkeling van het WWW. Samengevat zijn dat de volgende

- Geen beperkingen voor de gebruiker
- Eenvoudig aanmaken van koppelingen tussen documenten
- Decentralisatie
- Unieke adressering van documenten

Het eerste punt slaat op het besef dat iedereen op zijn eigen manier werkt. Om informatie met elkaar te delen moeten dus zo min mogelijk regels worden opgelegd, alleen een paar algemene regels rond de communicatie tussen computers. Hier zorgen TCP/IP, HTTP<sup>1</sup> en HTML<sup>2</sup> voor. Met HTTP en HTML werd hypertext geïntroduceerd in het WWW. Dit maakt het eenvoudig om koppelingen te maken tussen documenten.

De decentralisatie van het systeem is ook belangrijk. Zo kan een nieuwe persoon het systeem gebruiken zonder eerst toegang te hoeven vragen aan iemand anders. Doordat er geen centrale database is waarin alle koppelingen zijn opgeslagen, zou het nieuwe systeem goed te schalen zijn.

Elke knoop in het systeem moet met behulp van hypertext kunnen wijzen naar elke andere knoop, ofwel, elke knoop moet een uniek adres hebben. Dit is de Universal Resource Identifier (URI). Iedereen dwingen om HTTP te gebruiken, dat zou tegen het principe van minimale beperkingen zijn. Hoe kan het Web dan toch zo universeel mogelijk gehouden worden? De oplossing ligt in de URI. Als een document een URI heeft, kan het op een server neergezet worden en gevonden worden door een browser.

---

<sup>1</sup>Hypertext Transfer Protocol

<sup>2</sup>Hypertext Markup Language, een taal die beschrijft hoe pagina's met hyperlinks weergegeven moeten worden.

## Hoofdstuk 3

# Doel van het SW

Nadat het WWW snel gegroeid was, heeft Tim Berners-Lee het World Wide Web Consortium (W3C) opgericht. Doel daarvan is om algemene protocollen te ontwikkelen om de interoperabiliteit en ontwikkeling van het Web te versterken. Tegenwoordig speelt het W3C ook een grote rol in de ontwikkeling van het Semantische Web (SW). De Semantic Web Working Group (SWwg) van het W3C houdt zich hier mee bezig. Hun Activity Statement is:

The goal of the Semantic Web initiative is as broad as that of the Web: to create a universal medium for the exchange of data. It is envisaged to smoothly interconnect personal information management, enterprise application integration, and the global sharing of commercial, scientific and cultural data. Facilities to put machine-understandable data on the Web are quickly becoming a high priority for many organizations, individuals and communities [1, onder Activity].

We zien hier twee dingen naar voren komen:

- Universaliteit: “to create a universal medium...”
- Uitwisselbaarheid van informatie: “It is envisaged to smoothly interconnect personal information management, enterprise application integration and the global sharing of commercial, scientific and cultural data.”

Vanwege de universaliteit die het SW moet hebben, is een vorm van standaardisatie nodig. Iedereen moet het tenslotte erover eens zijn *hoe* informatie wordt uitgewisseld. Gebrekkige uitwisselbaarheid van informatie is (onder andere) één van de problemen van het WWW.

### 3.1 Problemen

Data op het WWW wordt gerepresenteerd in HTML. Deze taal is bedoeld om te beschrijven hoe de data weergegeven moet worden. De inhoud is echter gewoon in natuurlijke taal, zodat alleen mensen dat kunnen begrijpen. De tools begrijpen de HTML, maar niet de natuurlijke taal. Hierdoor ontstaan een aantal problemen:

1. *Zoeken naar informatie.* Een query levert bergen resultaten op, maar daar is maar een klein deel ook werkelijk nuttig. Daarbij moet de gebruiker die data ook nog eens doorzoeken om te kijken of die relevant is.
2. *Consistent houden van informatie.* Dit is gerelateerd aan het eerste punt. Het is moeilijk om informatie consistent te houden, er zijn ook nagenoeg geen tools voor. De webmaster van een bepaalde site moet dus zelf zorgen dat alle informatie consistent blijft.

3. *Uitwisselbaarheid van informatie.* Web-applicaties kunnen hun data moeilijk met elkaar uitwisselen. Er zijn bijvoorbeeld websites waar je kunt zoeken op aanbiedingen van verschillende aanbieders. Normaal gesproken wordt door die websites gebruik gemaakt van “wrapper-scripts” en heuristieken om de productinformatie van deze verschillende aanbieders uit hun webpagina’s te verkrijgen. Wordt de bronlayout echter aangepast, dan werkt dit niet meer, en moet het wrapper-script worden aangepast.

Oplossingen voor deze problemen vereisen dat alle of delen van de data op het WWW gestructureerd wordt, zodat die data voor machines begrijpelijk is. Tim Berners-Lee schrijft in [4, hoofdstuk 13] dat de volgende dingen daarvoor nodig zijn:

1. Talen ontwikkelen die meta-informatie voor documenten begrijpelijk maken voor machines. Daarnaast terminologiën (ook wel namespaces en ontologiën) ontwikkelen met behulp van deze talen en die op het Web beschikbaar maken.
2. Applicaties en architecturen ontwikkelen die zulke talen en terminologiën gebruiken om te helpen bij het vinden, benaderen, presenteren en onderhouden van informatiebronnen.

Wij zullen in deze scriptie alleen op het eerste punt in gaan. Er zijn al diverse technieken ontwikkeld die met algemene regels de gewenste informatie kunnen beschrijven door middel van typen, klassen etc. De belangrijkste zijn RDF<sup>1</sup>, RDFS<sup>2</sup> en OWL<sup>3</sup>. Queries op door deze technieken aangemaakte documenten kunnen uitgevoerd worden met de querytaal SPARQL<sup>4</sup>. Al deze technieken zijn op basis van XML<sup>5</sup>. In hoofdstuk 4 zullen we op RDF en RDFS ingaan. Op OWL en SPARQL zullen we verder niet ingaan.

Voordat we de technieken zullen bekijken, zullen we eerst wat meer ingaan op de vraag wat voor semantiek de technieken nu eigenlijk willen toekennen.

## 3.2 Semantiek in het SW

Waar zit nu precies de semantiek in het SW? Michael Uschold schrijft in [13] dat verschillende dingen semantiek hebben, zoals:

1. Termen of expressies die verwijzen naar dingen in de werkelijkheid, ‘real world subject matter’;
2. Termen of expressies in een taal die computers gebruiken om met elkaar te communiceren;
3. Een taal die bovenstaande informatie kan representeren.

Het derde punt ligt als het ware op metaniveau. Het beschrijft semantiek over termen of expressies. Op zijn beurt wordt er aan constructen in die taal ook een semantiek toegekend. Verder onderscheidt hij een soort ‘continuüm’ in het gebruik van het woord ‘semantiek’ [13]. Hij onderscheidt vier soorten, die gerangschikt kunnen worden van ambigu tot niet-ambigu.

1. Impliciet, wanneer mensen een soort gemeenschappelijke overeenstemming hebben over een bepaald begrip (bijvoorbeeld wat een woonadres is);
2. Expliciet en informeel, wanneer heel precies (in natuurlijke taal) wordt opgeschreven wat iets betekent (bijvoorbeeld: “de tag <h2> in HTML heeft de betekenis van een kop van grootte 2”);

---

<sup>1</sup>Resource Description Framework

<sup>2</sup>RDF Schema

<sup>3</sup>Web Ontology Language

<sup>4</sup>SPARQL Protocol And RDF Query Language

<sup>5</sup>Extensible Markup Language



3. Expliciet en formeel voor gebruik door mensen, wanneer de semantiek in een formele taal is vastgelegd en wel door mensen, maar niet direct door machines verwerkt kan worden;
4. Expliciet en formeel voor gebruik door machines, wanneer de semantiek in een formele taal is vastgelegd en verwerkbaar is door machines (bijvoorbeeld  $\langle x := x + 1, s_0 \rangle \rightarrow s[x \mapsto 1]$ , de uitvoering van het statement  $\langle x := x + 1, s_0 \rangle$  in toestand  $s_0$  resulteert in diezelfde toestand  $s_0$  behalve dan dat  $x$  de waarde 1 heeft).

Tussenvormen van deze soorten semantiek zijn natuurlijk ook mogelijk. Het verschil tussen de derde en vierde soort zit hem vooral in het gebruik ervan. Waar mensen met een formele taal nog wel eens fouten maken, doen computers dat niet (tenzij ze niet goed geprogrammeerd zijn). In andere literatuur [12] wordt er overigens een ander onderscheid gemaakt, waar we hier verder niet op in zullen gaan.

Het doel van het SW is nu dat machines weten wat ze met data moeten doen die ze tegenkomen. Hiervoor is het nodig dat de data '4-semantiek' heeft. Niet alle data kan gevat worden in 4-semantiek. Wel kan je iets zeggen over Dingen die 1-, 2- of 3-semantiek hebben. (Denk hierbij bijvoorbeeld aan een adres, deze heeft een impliciete semantiek voor ons mensen. Expliciet en formeel zou dat uitgedrukt kunnen worden in een straatnaam en nummer, die dan weer beide een type hebben etc.).

In het volgende hoofdstuk zullen we ingaan hoe RDF en RDFS werken. Beiden zijn voorbeelden van 4-semantiek; RDF(S)-documenten kunnen door machines worden geïnterpreteerd. Ook de makers van RDF(S) zien in dat deze techniek maar een bepaalde vorm van semantiek toekent, in RDF-semantics [8] staat het volgende over de semantiek die RDF toekent:

The semantics given here restricts itself to a formal notion of meaning which could be characterized as the part that is common to all other accounts of meaning, and can be captured in mechanical inference rules.

...

The chief utility of a formal semantic theory is not to provide any deep analysis of the nature of the things being described by the language or to suggest any particular processing model, but rather to provide a technical way to determine when inference processes are valid, i.e. when they preserve truth.

RDF is dus bedoeld om Dingen 4-semantiek toe te kennen. Deze semantiek zegt echter niet alles over een bepaald Ding. Wat je met 4-semantiek zegt over dat Ding, blijft echter ook waar in 1-, 2- en 3-semantiek.

# Hoofdstuk 4

## Technieken

In dit hoofdstuk zullen we de verschillende technieken bespreken die gebruikt worden voor het Semantische Web. Alle technieken zijn gebaseerd op XML. Deze techniek is bedoeld om data te beschrijven. We zullen eerst kort XML en XML Namespaces bespreken. Vervolgens komen RDF en RDFS iets uitgebreider aan bod.

### 4.1 XML

In figuur 4.1 staat een voorbeeld van een XML-document.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note date="12/11/2002">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Figuur 4.1: XML

In een XML-document zijn de volgende termen van belang:

- Root-element
- Element
- Attribuut

De XML declaratie (`<?xml version="1.0" encoding="ISO-8859-1"?>`) is het begin van ieder XML-document. Hierin staat de XML-versie en de karakterset van het document. Verder is het document opgebouwd uit elementen (onderscheiden door start- en sluittags, zoals `<to>` en `</to>`), al dan niet vergezeld van een attribuut. Deze elementen mag de auteur van een document zelf bedenken (vandaar de X van eXtensible). Het document moet één root-element hebben, hier is dat `<note>`. De elementen zijn hier `<note>`, `<to>`, `<from>`, etc. Een voorbeeld van een attribuut is `date="12/11/2002"` binnen de `<note>` tag.

### 4.2 XML Namespaces

Wanneer je nu twee XML-documenten wilt samenvoegen, kun je tegen het probleem aanlopen dat die twee documenten elementen bevatten die beide een zelfde naam hebben, maar die niet hetzelfde

bedoelen te beschrijven. Met XML Namespaces kan dit worden opgelost. Zij zijn ontworpen om universele, unieke namen te geven voor elementen en attributen in XML-documenten. Zo kunnen twee XML documenten (of delen daarvan) zonder conflicten in naamgeving samengevoegd worden. Dit zorgt ervoor dat iedereen zijn eigen namen voor elementen en attributen kan geven.

Stel dat iemand een document heeft gemaakt dat een (woon)adres representeert:

```
<?xml version="1.0" ?>
<Address>
  <Street>Wilhelminenstr. 7</Street>
  <City>Darmstadt</City>
  <State>Hessen</State>
  <Country>Germany</Country>
  <PostalCode>D-64285</PostalCode>
</Address>
```

Figuur 4.2: XML - woonadres

Tegelijk kan diegene een server representeren die een bepaald IP-adres heeft:

```
<?xml version="1.0" ?>
<Server>
  <Name>OurWebServer</Name>
  <Address>123.45.67.8</Address>
</Server>
```

Figuur 4.3: XML - serveradres

Beide documenten bevatten nu een tag `<Address>`, hierdoor is er een ambiguïteit. De twee documenten kunnen zonder ambiguïteiten worden samengevoegd door gebruik te maken van XML Namespaces. Een departement dat gevestigd is op een bepaald adres en een server heeft met een bepaald IP-adres, kan nu zoals in figuur 4.4 worden weergegeven. Er wordt voor ieder element binnen een bepaalde namespace een voorvoegsel gezet. Het straatadres heeft hier het voorvoegsel `addr` en het serveradres het voorvoegsel `serv`. Deze worden opgegeven door een `xmlns`-attribuut in het omvattende element (hier is dat `xmlns:addr="http://www.tu-darmstadt.de/ito/addresses` en `xmlns:serv="http://www.tu-darmstadt.de/ito/servers"` in het `<addr:Address>`- en `<serv:Server>`-element.

## 4.3 RDF

De eerste W3C recommendation over RDF (Resource Description Framework) is van 1999. De nieuwste stamt uit 2004. Het is bedoeld om informatie mee te beschrijven. Deze metadata kan daarna uitgewisseld worden tussen verschillende systemen. Hierbij is het doel in eerste plaats interpretatie en uitwisseling door en tussen machines. Op die manier kunnen machines automatisch (zonder tussenkomst van mensen) met data redeneren.

RDF-documenten zijn eigenlijk gespecialiseerde XML-documenten. De tags hebben hier echter een formeel gedefinieerde betekenis.

Met RDF kun je statements maken over resources. Deze statements bestaan uit zgn. drietallen waarmee een graaf wordt opgebouwd. De drietallen bestaan uit:

- Resource
- Property

```

<?xml version="1.0" ?>
<Department>
  <Name>DVS1</Name>

  <addr:Address xmlns:addr="http://www.tu-darmstadt.de/ito/addresses">
    <addr:Street>Wilhelminenstr. 7</addr:Street>
    <addr:City>Darmstadt</addr:City>
    <addr:State>Hessen</addr:State>
    <addr:Country>Germany</addr:Country>
    <addr:PostalCode>D-64285</addr:PostalCode>
  </addr:Address>

  <serv:Server xmlns:serv="http://www.tu-darmstadt.de/ito/servers">
    <serv:Name>OurWebServer</serv:Name>
    <serv:Address>123.45.67.8</serv:Address>
  </serv:Server>

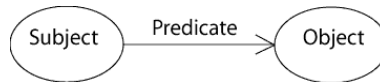
</Department>

```

Figuur 4.4: XML - samengevoegd document

- Property value

Deze drie termen worden ook wel resp. subject, predicate en object genoemd. Grafisch wordt dit als volgt weergegeven:



Figuur 4.5: RDF - datamodel

Kort gezegd definieert RDF de notatie en het datamodel voor machine-begrijpelijke data. Om bovenstaande algemene begrippen te verduidelijken zullen we een voorbeeld bespreken.

Stel dat je een lijst met gegevens over CD's hebt<sup>1</sup> zoals in tabel 4.1.

| Title            | Artist       | Country | Company     | Price | Year |
|------------------|--------------|---------|-------------|-------|------|
| Empire Burlesque | Bob Dylan    | USA     | Columbia    | 10.90 | 1985 |
| Hide your heart  | Bonnie Tyler | UK      | CBS Records | 9.90  | 1988 |
| ...              |              |         |             |       |      |

Tabel 4.1: CD-tabel

Dan kun je dat met RDF zoals in figuur 4.6 weergeven. De eerste regel is de XML-declaratie, die aangeeft dat het om een XML-document gaat. Daarna volgt een `<rdf:RDF>` element, dit is het root-element van het RDF-document. Het geeft aan dat dit XML-document een RDF document is. Er worden twee namespaces gedeclareerd om ambiguïteiten te vermijden.

Daarna volgt het `<rdf:Description>` element. Deze bevat de beschrijving van de *resource* die gespecificeerd wordt door het `rdf:about` attribuut. De elementen `<cd:artist>`, `<cd:country>`, `<cd:company>` etc. beschrijven de betreffende resource zoals in de tabel weergegeven. De tags zijn de *property* (eigenschap)-namen, de stukjes tekst tussen de start- en sluittags zijn de *property-values* (waarden van de eigenschap). Merk op dat `<cd:name>` ook opgenomen is. Weliswaar zou

<sup>1</sup>Voorbeeld overgenomen van [http://www.w3schools.com/rdf/rdf\\_example.asp](http://www.w3schools.com/rdf/rdf_example.asp)

```

<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
    <cd:name>Empire Burlesque</cd:name>
    <cd:artist>Bob Dylan</cd:artist>
    <cd:country>USA</cd:country>
    <cd:company>Columbia</cd:company>
    <cd:price>10.90</cd:price>
    <cd:year>1985</cd:year>
  </rdf:Description>

  <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Hide your heart">
    <cd:name>Hide your heart</cd:name>
    <cd:artist>Bonnie Tyler</cd:artist>
    <cd:country>UK</cd:country>
    <cd:company>CBS Records</cd:company>
    <cd:price>9.90</cd:price>
    <cd:year>1988</cd:year>
  </rdf:Description>
  .
  .
  .
</rdf:RDF>

```

Figuur 4.6: RDF

de CD-naam ook afgeleid kunnen worden uit het `rdf:about` attribuut, maar dit attribuut is echter alleen bedoeld om resources unieke namen te kunnen geven (door een URI). De waarde van het attribuut `rdf:about` heeft dus geen speciale betekenis.

Naast het `<RDF>` en `<Description>` element zijn er nog elementen die groepen van dingen beschrijven. Dit zijn de Container-tags: `<Bag>`, `<Seq>` en `<Alt>`. De Bag-tag beschrijft een ongeordende verzameling resources, een Seq-tag een geordende. De Alt-tag beschrijft een lijst met alternatieve waarden. De elementen binnen deze container-tags worden *members* genoemd. Een voorbeeld van een container-tag staat in figuur 4.7. Daarin staan de members binnen de Bag-tag, ieder gescheiden door een `<rdf:li>` element.

Een container beschrijft dat de resources binnen die container members zijn. Het zegt echter *niet* dat andere resources niet toegestaan zijn. Hiervoor is het Collection-attribuut bedoeld. Deze beschrijft een groep die alleen de members mag bevatten die hierin worden opgegeven. Een Collection wordt aangegeven door het attribuut `rdf:ParseType="Collection"`, zie figuur 4.8.

## 4.4 RDF Schema

RDF heeft vrij weinig uitdrukingskracht. Je kunt er alleen basisbeschrijvingen mee geven. RDF Schema is een uitbreiding van het datamodel van RDF. Hierin worden de volgende dingen toegevoegd:

- klassen

```

<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Beatles">
<cd:artist>
  <rdf:Bag>
    <rdf:li>John</rdf:li>
    <rdf:li>Paul</rdf:li>
    <rdf:li>George</rdf:li>
    <rdf:li>Ringo</rdf:li>
  </rdf:Bag>
</cd:artist>
</rdf:Description>

</rdf:RDF>

```

Figuur 4.7: RDF - Container

```

<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://recshop.fake/cd#">

<rdf:Description
rdf:about="http://recshop.fake/cd/Beatles">
<cd:artist rdf:parseType="Collection">
  <rdf:Description rdf:about="http://recshop.fake/cd/Beatles/George"/>
  <rdf:Description rdf:about="http://recshop.fake/cd/Beatles/John"/>
  <rdf:Description rdf:about="http://recshop.fake/cd/Beatles/Paul"/>
  <rdf:Description rdf:about="http://recshop.fake/cd/Beatles/Ringo"/>
</cd:artist>
</rdf:Description>

</rdf:RDF>

```

Figuur 4.8: RDF - Collection

- overerving van klassen en eigenschappen
- domein- en bereikrestricties voor eigenschappen

Hiermee kan meer uitgedrukt worden dan in RDF. De bedoeling is dat gebruikersgemeenschappen zelf hun *voculaires* zullen gaan beschrijven met RDFS<sup>2</sup>. Bijvoorbeeld: mensen die bibliografische resources willen beschrijven, zullen de klassen Boek en TijdschriftArtikel en eigenschappen auteur, titel en onderwerp willen beschrijven. Dit kan niet in RDF. Met RDFS worden dan niet deze specifieke klassen en eigenschappen van dit applicatiedomein beschreven, maar dit soort klassen en eigenschappen kunnen er wel mee worden beschreven. RDFS zorgt dus als het ware voor een soort typesysteem voor RDF. Hiermee worden hiërarchiën van klassen gemaakt, en resources beschreven als instanties van één of meer klassen.

Klassen worden beschreven met `rdfs:Class`, `rdfs:Resource`, `rdf:type` en `rdfs:subClassOf`. In figuur 4.9 geven we een voorbeeld van het gebruik van RDFS. Hierin worden een aantal klassen beschreven: 'MotorVehicle', 'Van', en 'Person'. Hierin is

```
<rdfs:Class rdf:ID="MotorVehicle"/>
```

een verkorte schrijfwijze voor:

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

Bij de klasse 'Van' is de eigenschap 'subClassOf' opgenomen. De waarde hiervan is '#MotorVehicle', deze verwijst naar de klasse 'MotorVehicle'. Een machine kan uit deze (RDFS)eigenschap afleiden dat een 'Van' een 'MotorVehicle' is.

`rdfs:Property`, `rdfs:domain`, `rdfs:range` en `rdfs:subPropertyOf` worden gebruikt om RDF-properties mee te beschrijven. In figuur 4.9 zijn drie eigenschappen beschreven: 'registeredTo', 'driver' en 'primaryDriver'. Met `rdfs:range` kan worden aangegeven dat waarden van een bepaalde eigenschap instanties zijn van een bepaalde klasse. (Het specificeert dus de 'object'-kant van een RDF-drietal, zie figuur 4.5) In het voorbeeld geeft `<rdfs:range rdf:resource="#Person"/>` bij de eigenschap 'registeredTo' bijvoorbeeld aan dat iets alleen 'registeredTo' een 'Person' kan zijn.

`rdfs:domain` geeft juist aan dat een bepaalde eigenschap toegepast wordt op een bepaalde klasse. (Het specificeert dus de 'subject'-kant van een RDF-drietal, zie figuur 4.5). De eigenschap 'registeredTo' wordt dus toegepast op instanties van de klasse 'MotorVehicle'.

Net als bij klassen kunnen ook eigenschappen subeigenschappen zijn van andere eigenschappen. In het voorbeeld is 'primaryDriver' een subeigenschap van 'driver'.

---

<sup>2</sup>Vandaar dat RDFS officieel RDF Vocabulary Description Language heet.

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdfs:Class rdf:ID="MotorVehicle"/>

  <rdfs:Class rdf:ID="Van">
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdfs:Class>

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

  <rdf:Property rdf:ID="registeredTo">
    <rdfs:domain rdf:resource="#MotorVehicle"/>
    <rdfs:range rdf:resource="#Person"/>
  </rdf:Property>

  <rdf:Property rdf:ID="driver">
    <rdfs:domain rdf:resource="#MotorVehicle"/>
  </rdf:Property>

  <rdf:Property rdf:ID="primaryDriver">
    <rdfs:subPropertyOf rdf:resource="#driver"/>
  </rdf:Property>

</rdf:RDF>

```

Figuur 4.9: RDFS



## Hoofdstuk 5

# Conclusie

We zijn in deze scriptie op zoek gegaan naar wat het Semantische Web nu eigenlijk is. Daarbij begonnen we met zoeken door in te gaan op het ontstaan van het World Wide Web. In het begin van het WWW waren er al gedachten over machines die zelfstandig koppelingen tussen data kunnen nagaan, analyseren en vervolgens iets zinvolsv daarmee doen. Data moet dus begrijpelijk zijn voor machines.

Er is al ontzettend veel informatie gekoppeld in het Web. Voor machines valt uit die koppelingen echter weinig méér op te maken dan dat de betreffende data aan elkaar gekoppeld is. Wat die koppelingen inhouden is wel bekend voor mensen, maar niet voor machines. Hierdoor ontstaan een aantal problemen die vooral verband houdt met de hoeveelheid informatie. Is er weinig informatie om te doorzoeken of bij te houden, dan kunnen mensen dat uitstekend zelf. Wordt die informatie echter omvangrijk (zoals bij het Web het geval is), dan wordt het een probleem om informatie te zoeken. Het kost te veel tijd om alle informatie handmatig door te gaan om te kijken of die relevant is. Ook het bijhouden van een omvangrijke website wordt, om dezelfde reden, erg moeilijk.

Wanneer data voor machines begrijpelijk is, kunnen zij zelf uitmaken of een document relevant is of dat een document consistent is met andere documenten. Ofwel, door de data zó te structureren dat het een voor machines begrijpelijke semantiek heeft, kunnen machines de data op een zinvolle manier verwerken. Maar wanneer is die semantiek voor machines begrijpelijk, en over wat voor soort semantiek gaat het hier? We hebben gezien dat een expliciete formele semantiek nodig is om voor machines begrijpelijk te zijn.

De technieken RDF en RDFS zijn bedoeld om data mee te beschrijven. De beschrijving die RDF geeft, is in wezen een gerichte graaf. Hierbij stelt de pijl een bepaalde eigenschap voor, de knoop waar de pijl vandaan komt is de resource waar die eigenschap voor geldt en de knoop waar de pijl naar toe wijst is de waarde van die eigenschap. Met RDFS kan meer worden beschreven. Hiermee kan worden beschreven dat een bepaalde resource een subklasse is van een andere resource. Ook kunnen domein en bereik restricties worden beschreven op eigenschappen of subeigenschappen.

Met RDFS kunnen dus terminologiën gemaakt worden. Deze geven een voor machines begrijpelijke semantiek, maar zijn verweven met betekenis die mensen aan bepaalde dingen geven (impliciete semantiek). Deze twee semantieken zijn niet gelijk, maar de voor machines begrijpelijke semantiek blijft haar (logische) waarheid wel behouden in de impliciete semantiek van mensen.

Het Semantische Web draait om machine-begrijpelijke data. Dit kan met RDF, RDFS en extensies daarvan beetje bij beetje verwerkelijkt worden. Niet alle data zal begrijpelijk worden voor machines, toch zullen zij door deze technieken beter met data kunnen omgaan. Vinden, consistent houden en uitwisselen van informatie wordt dus meer een zaak van machines, en minder van mensen.

# Bibliografie

- [1] W3C Semantic Web homepage, <http://www.w3.org/2001/sw/>
- [2] Dave Beckett (editor), *RDF/XML Syntax Specification (Revised)*, 2004
- [3] J.F.A.K. van Benthem, H.P. van Ditmarsch, J. Ketting en W.P.M. Meyer-Viol, *Logica voor informatici*, Addison-Wesley Nederland, tweede druk 1993
- [4] Tim Berners-Lee, *Weaving the Web*, HarperCollins Publishers, 1999
- [5] Dan Brickley en R.V. Guha (editors), *RDF Vocabulary Description Language 1.0: RDF Schema*, 2004
- [6] Vannevar Bush, *As We May Think*, The Atlantic Monthly, juli 1945
- [7] Dieter Fensel, James Hendler, Henry Lieberman en Wolfgang Wahlster (editors), *Spinning the Semantic Web*, The MIT Press, 2003
- [8] Patrick Hayes (editor), *RDF Semantics*, <http://www.w3.org/TR/rdf-mt/>, 2004
- [9] Graham Klyne en Jeremy J. Carroll (editors), *Resource Description Framework (RDF): Concepts and Abstract Syntax*, <http://www.w3.org/TR/rdf-concepts/>, 2004
- [10] Ted Nelson, *Literary Machines*, Mindful Press, 1981
- [11] Hanne Riis Nielson en Flemming Nielson, *Semantics with applications - a formal introduction*, (original edition John Wiley & Sons, 1992), 1999
- [12] Amit Sheth, Cartic Ramakrishnan and Christopher Thomas, *Semantics for the Semantic Web: The Implicit, the Formal and the Powerful*, Intl Journal on Semantic Web & Information Systems, 1(1), 1-18, Jan-March 2005
- [13] Michael Uschold, *Where are the Semantics in the Semantic Web*, AI Magazine, 2003