Linked Data en semantische technologie voor data-integratie en mashups

Bouwstenen van het semantisch web

Paul Hermans

Er bestaat grote verwarring over wat het semantisch web is en wat het inhoudt. Dit artikel zet één en ander op een rij en probeert ook aan te duiden wat de relevantie van deze technologie is voor en binnen het bedrijf.

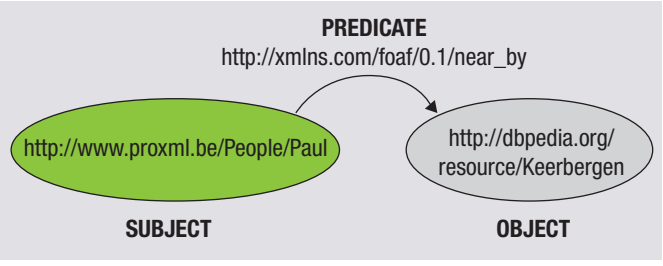
We beginnen met de basis bouwblokken van het semantisch web om dan via de Linked Data beweging, waarbij weinig semantiek komt kijken, te eindigen bij de wereld van description logics en het gebruik van *full-blown ontology's*.

RDF: het datamodel voor het web

Naast documenten moesten er volgens het World Wide Web Consortium ook data op het web komen. Het web heeft echter specifieke kenmerken. Zo is het web fundamenteel een decentraal gebeuren. Op elk moment kan iedereen een statement doen, c.q. data poneren over welk onderwerp dan ook. Dit noemt men het AAA-principe: *Anyone can say Anything about Any topic*. Verder mag men ervan uitgaan dat men nooit over alle informatie beschikt; op ieder moment kunnen nieuwe informatie of nieuwe data opduiken. Dit is de *Open World Assumption*. Dit uitgangspunt heeft allerlei onverwachte consequenties, voor als we gaan modelleren en redeneren.

Omdat iedereen uitspraken kan doen over ieder onderwerp betekent dit ook dat iedereen een verschillende *identifier* kan geven aan dat onderwerp: de *Non Unique Naming Assumption*. Dezelfde entiteit, hetzelfde ding kan dus meerdere unieke identifiers hebben op het web.

Het gebruikte datamodel, *RDF* (Resource Description Framework) genoemd, moest deze principes dan ook reflecteren. RDF is bijgevolg schemaloos en iedereen kan relaties leggen tussen paren van resources (of een resource en een atomaire datavalue).



Afbeelding 1: Triple.

Een voorbeeld van zo'n statement van een relatie tussen twee resources:

- een persoon met unieke identifier, bijvoorbeeld http://www.proxml.be/People/Paul;
- een gemeente met unieke identifier, bijvoorbeeld http://dbpedia.org/resource/Keerbergen.

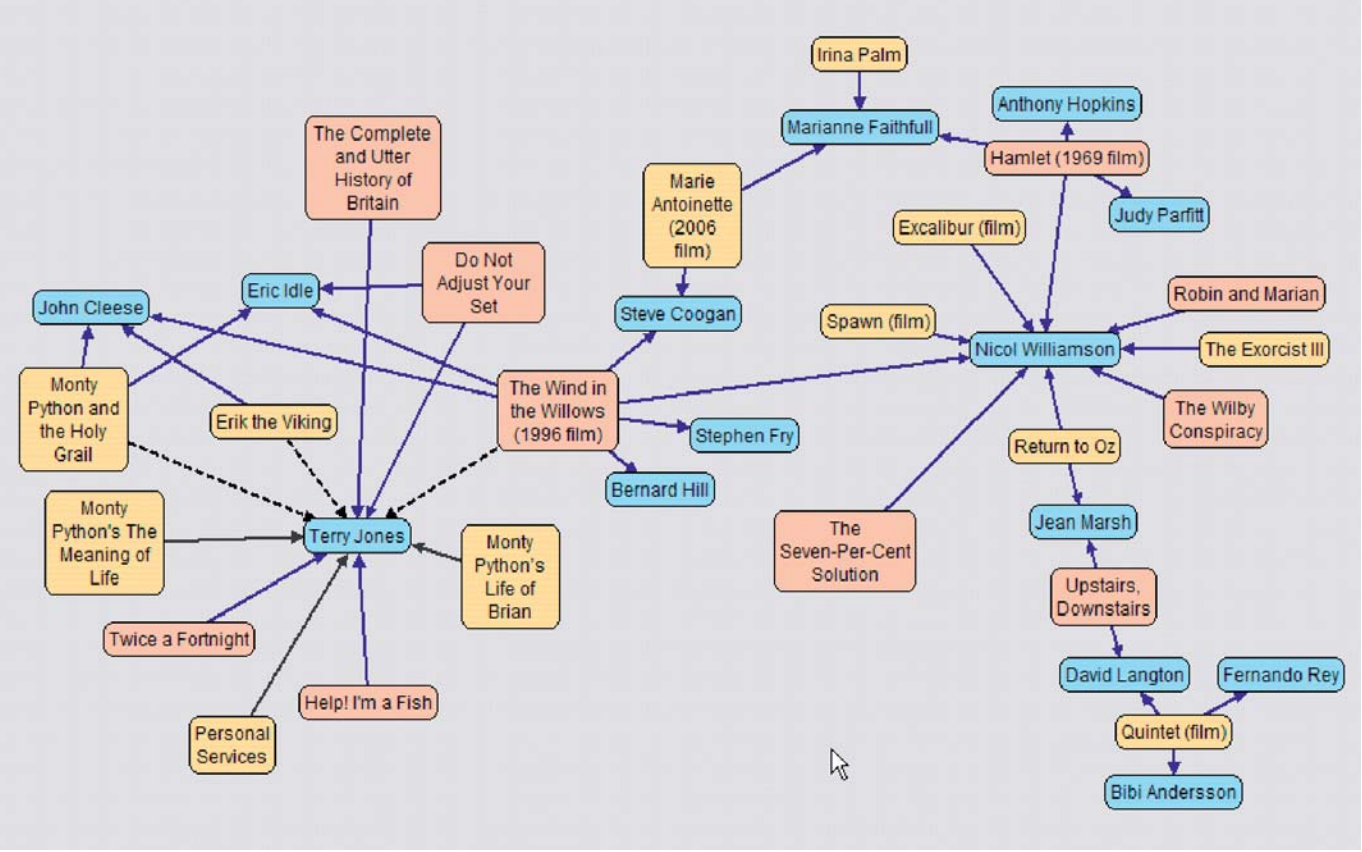
Een benoemde relatie, namelijk http://xmlns.com/foaf/0.1/near_by tussen die twee. Dit noemt men een *triple*. Zo'n triple bestaat uit een subject, een predicate en een object, zie afbeelding 1. Vele triples samen vormen een *graph*, zie afbeelding 2.

Voordeel

Wat is nu het grote voordeel van dit graph datamodel? Welnu, het maakt het mergen van verschillende datasets waarin gemeenschappelijke identifiers gebruikt zijn triviaal. En hierin schuilt ook één van de 'selling points' van semantische technologie binnen het bedrijf, namelijk het gemak waarmee deze data-integratie toelaat. Een aandachtige lezer zal opwerpen; wat indien mijn data geen gemeenschappelijke identifiers hebben? Dan moeten we meer trucs uit de semantische webdoos halen, maar daarover verderop meer bij het ontologisch modelleren.

Hoe komt het dat ondanks deze kracht RDF tot voor kort niet zo populair was? Dit kwam doordat RDF initieel in RDF/XML werd geserialiseerd in een zeer overladen en bijgevolg zo goed als onleesbare syntax. Gelukkig nemen andere serialisaties zoals N3 en Turtle ondertussen de overhand.

U hoeft zich daar niet te veel van aan te trekken want er zijn ondertussen voldoende library's die al deze serialisatieformaten onderling kunnen converteren. Maar, 'to set the record straight':



Afbeelding 2: Graph.

RDF is geen XML. Er bestaan heel wat tools en library's om legacy data en formaten zoals RDBM's tabellen en views, Excel spreadsheets, CSV enzovoort om te zetten naar RDF. Het is niet zo moeilijk om RDF aan te maken.

SPARQL, de bijhorende query taal

Voor query's op RDF data is SPARQL ontworpen. SPARQL Protocol and RDF Query Language bestaat uit twee delen:

- een SQL-vergelijkbare taal voor het bevragen van sets van RDF graphs;
- een protocol voor het stellen van query's en het opvragen van resultaten over HTTP.

Zie afbeelding 3 voor een voorbeeld van een SPARQL query. Hiermee is de basis van het semantisch web gelegd. Er is een datamodel (RDF) dat toelaat om decentraal data te creëren, deze data dan vlot te mergen en deze dan op een standaard manier te bevragen (SPARQL).

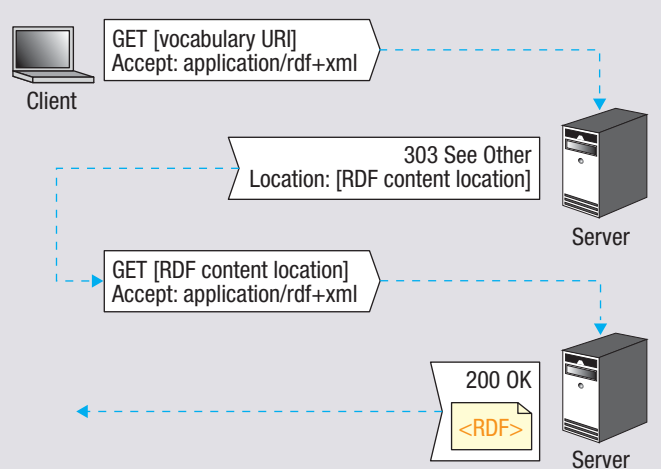
Linked Data

Wij kennen het web als een web van documenten. Het geprefereerde formaat van webdocumenten is HTML. Elk document heeft een URL, waarmee het kan worden opgevraagd en documenten zijn onderling verbonden met hyperlinks.
Ditzelfde wilde men ook voor data. Het hebben van RDF alleen leidde niet echt tot het gebruik ervan op het web. Daarvoor ontbrak nog een aantal stukjes van de puzzel. Daarom stelde Tim Berbers-Lee in 2006 de volgende richtlijnen voor:

- Gebruik URI's als identifier van entiteiten/dingen zoals personen, organisaties, boeken, genen enzovoort;
- Gebruik HTTP URI's zodat deze kunnen worden opgevraagd, met andere woorden gebruik 'dereferenceable' URL's, zijnde URL's die leiden naar representaties van entiteiten (HTML, XML, RDF);
- Als een URI wordt opgevraagd, zorg dan dat er zinvolle informatie wordt gegeven, gebruik makend van een standaard, zijnde RDF;

```
SELECT?person?collegeOfSpouse
WHERE{
      ?person :gender :male.
      ?person :birthYear ?yearOfBirth.
      ?person :spouse ?spouse.
      ?spouse :almaMater ?collegeOfSpouse.
FILTER (?yearOfBirth < 1950)|
}
ORDER BY ?collegeOfSpouse
LIMIT 5
```

Afbeelding 3: SPARQL query.



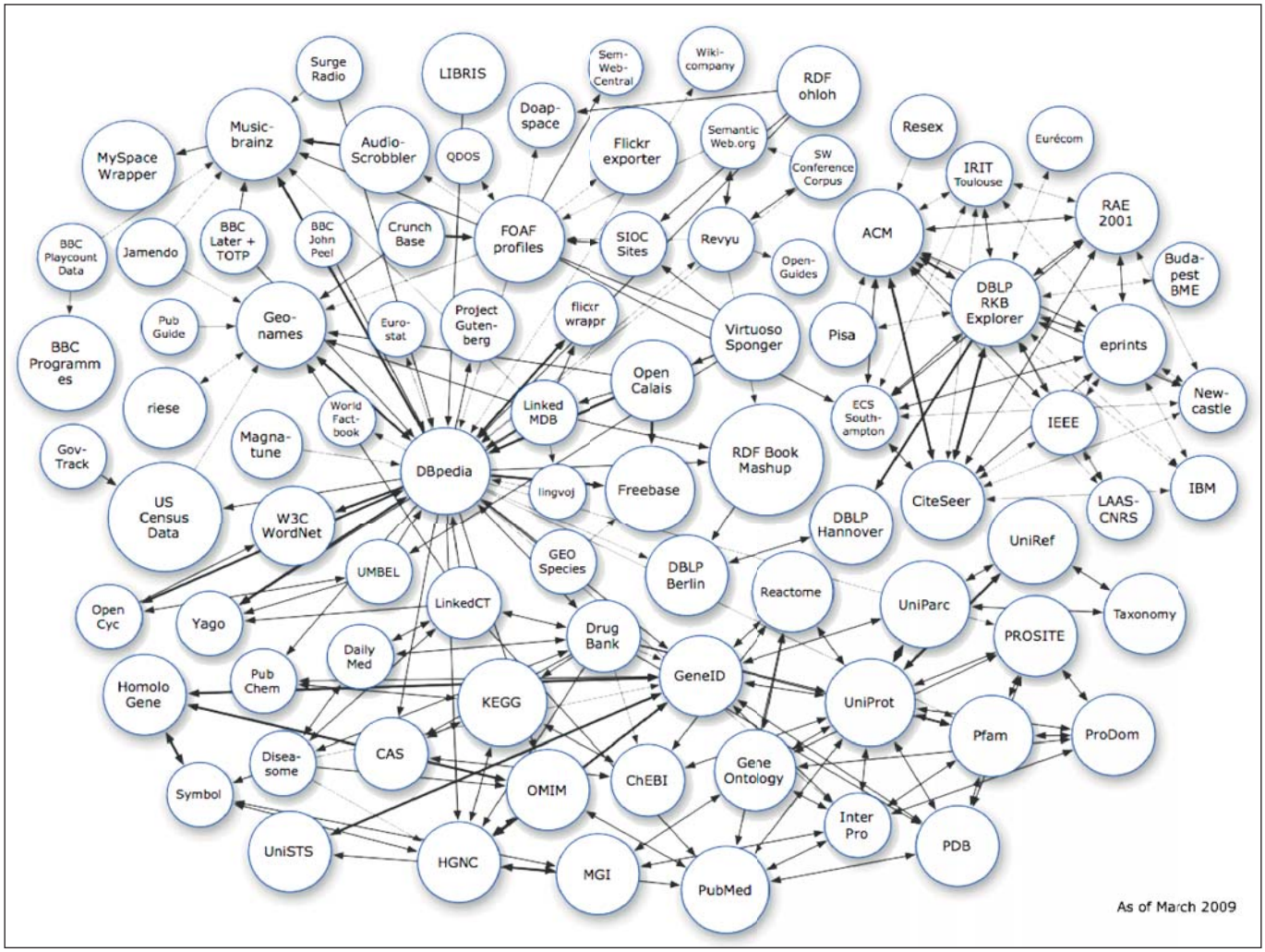
Afbeelding 4: Redirect.

- Zorg dat er in die informatie linken staan naar andere URI's zodat er meer data kunnen ontdekt worden; het 'Follow your nose' principe, identiek aan het volgen van hypertext linken.

Hoe maak je URL's die iets identificeren dereferenceable? Een persoon heeft een unieke identifier gekregen, maar die persoon zelf kan natuurlijk niet op het web gezet worden; het zijn de desbetreffende informatieresources (RDF statements) die op het web kunnen. Men moet een mechanisme hebben dat bij het ingeven van de identifier van een non-informatie resource, zoals een persoon, de relevante informatie resource (RDF document) teruggeeft. Er zijn hier meerdere technieken mogelijk, maar een veelgebruikte is met behulp van HTTP *content negotiatie*.
Ik doe een HTTP GET voor mijn identifier http://www.proxml.be/People/Paul, mijn web client accepteert MIME-type application/rdf+xml; de web server zal dan een redirect doen naar bijvoorbeeld http://www.proxml.be/People/Paul.rdf. Zie afbeelding 4.

Linken

Momenteel zijn er al immens veel datasets als linked data gepubliceerd. Dit is het gevolg van het Linking Open Data community project. Midden 2009 zijn er meer dan 100 open data sets met meer dan 4,7 biljoen RDF triples en onderling verbonden door ongeveer 142 miljoen links, zie afbeelding 5.
Hoe vind ik nu de unieke identifiers van de entiteiten waarnaar ik wil linken? Er bestaan ondertussen meerdere lookup-services zoals Sindice en Falcons, die daarvoor gebruikt kunnen worden.



Afbeelding 5:Voorbeeld onderling verbonden data sets.

Omschrijving	RDFS/OWL Statement
Klasse met id "Organisatie"	:Organisatie rdf:type owl:Class
Klasse met id "OverheidsOrganisatie"	:OverheidsOrganisatie rdf:type owl:Class
"OverheidsOrganisatie" is een subklasse van "Organisatie"	:OverheidsOrganisatie rdfs::subClassOf :Organisatie

Afbeelding 6: Ontologische statements.

Bijvoorbeeld, ik heb met Falcons gezocht op de string 'John Zorn'. De componist en saxofonist met naam 'John Zorn' heeft onder andere (denk terug aan de Non Unique Naming Assumption) als unieke identifier 'http://dbpedia.org/resource/John_Zorn'. Deze identifier kan ik nu gebruiken als link. Natuurlijk schaalt deze aanpak niet echt; vandaar dat er automatische linkgeneratie hulpmiddelen zoals Silk (http://www4.wiwiss.fu-berlin.de/bizer/silk/) komen.

Linked data browsers en search engines

Linked data kunnen ook gebrowsed worden volgens het 'Follow your nose' principe, identiek aan het browsen van webpagina's. Er is een Firefox plug-in met de naam Tabulator om RDF te exploreren. Verder zijn er Disco, Marbles, Openlink RDF Browser, Zitgist, ObjectViewer. Keuze genoeg, maar toch is dit nog steeds een pijnpunt van de linked data wereld dat deze browsers te zeer gericht zijn op de al technisch onderlegde gebruiker. Ik moet nog de eerste linked data browser zien die door het grote publiek moeiteloos kan gebruikt worden.
Er zijn enkele search engines ontwikkeld die door de linked data cloud *crawlen* door RDF-links te volgen: SWSE, Falcons, Swoogle, Sindice, Watson.

Verhouding Linked Data en Web 2.0 API's

Bij Web 2.0 API's gaat het ook om data. Elke service definieert echter zijn eigen API('s), daarbij kiezend uit verschillende families zoals SOAP, XML-RPC en REST, waarbij de dataset gebonden is aan de service. Het resultaatformaat is XML (SOAP of andere), JSON enzovoort. Dit betekent een grote variabiliteit. Hoe meer datasets en API's, hoe meer 'loodgieterij' er nodig is om een Mashup te maken. Linked data daarentegen gebruiken slechts één mechanisme: URI's, dereferenceable over HTTP en RDF als formaat. "Any data, one API." De scoop van de data is bovendien ongebonden: het gaat om één globale dataspace.
Een voorbeeld van een linked data-applicatie is de Music site van de BBC (www.bbc.co.uk/music/); een site gebouwd gebruikmakend van allerlei eigen BBC data gekoppeld aan twee publieke data sets: Musicbrainz en DBPedia.

Semantiek in de Linked Data aanpak

Er zit alleen semantiek in het property deel van de data triples. Property's hebben meer dan een label; zij hebben een unieke identifier. In het semantische web datamodel zijn property's *first class citizens*. Property's staan op zichzelf. Daarin verschilt dit van het traditionele OO- of RDBMS-denken waarin property's bestaan binnen de klasse of de tabel.
Er wordt aangeraden om hiervoor zoveel mogelijk gebruik te maken van gekende genoemde relaties. Zo hebben wij in ons eerste voorbeeld als relatie 'http://xmlns.com/foaf/0.1/near_by' gebruikt. Dit is een property uit het bekende FOAF (FriendOfAFriend) vocabularium dat property's definieert om personen, de linken tussen hen en de dingen die zij maken en doen te beschrijven. Dit vocabularium is goed gedocumenteerd zodat voor iedereen duidelijk is wat de precieze semantiek van zo'n property is.

Meer semantiek: ontologieën

Wanneer we een domein beschrijven dan gaan we niet alleen individuen en hun relaties beschrijven maar ook hun types of classes zoals 'persoon', 'organisatie', 'overheidsorganisatie'. Wij mensen weten dat elke overheidsorganisatie een organisatie is en dat personen hiervoor kunnen werken. De vraag is hoe deze kennis zo formeel mogelijk vast te leggen, zodoende dat ook machines hiermee kunnen werken.

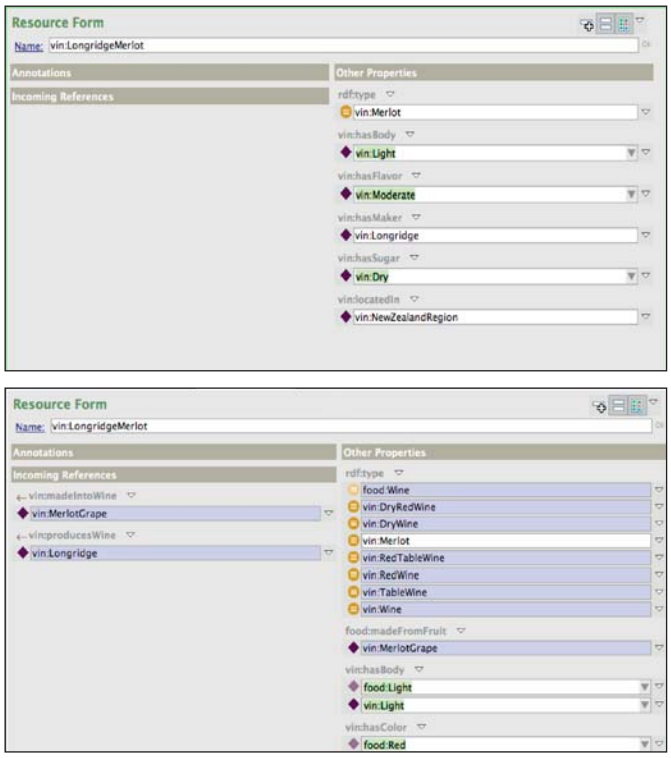
Voor gebruik op het web zijn hiervoor enkele ontology-talen ontwikkeld: RDFS en OWL in zijn verschillende versies, subtalen en profielen. Deze varianten zijn er omdat expressiviteit een kostenaspect heeft. Hoe expressiever (hoe meer men formeel kan vastleggen) de ontology-taal, hoe moeilijker het is om software te maken die binnen een redelijke tijd, juiste en volledige afleidingen kan maken. De OWL-FULL soort die het meest expressief is, is zelfs 'undecidable'. Enkele voorbeelden van ontologische statements staan in de tabel in afbeelding 6. Deze RDFS/OWL statements zijn ook RDF triples, wat betekent dat data statements en model statements hetzelfde datamodel volgen en dus geïntegreerd gebruikt kunnen worden met dezelfde tools en querytalen. Dit is een erg groot voordeel.
Wat is nu precies de semantiek van deze ontologische statements? De semantiek is geformaliseerd door te specifiëren welke 'inferences', logische gevolgtrekkingen er op basis van een

Statement	Inference
:CBVS a :OverheidsOrganisatie :OverheidsOrganisatie rdfs::subClassOf :Organisatie	:CBVS a :Organisatie

Afbeelding 7: Voorbeeld.

Statement	Inference
:ligtIn a owl:TransitiveProperty :A :ligtIn :B :B :ligtIn :C	:A :ligtIn :C

Afbeelding 8: Voorbeeld.



Afbeelding 9: Boven, zonder inferences. Onder, met inferences.

ontologische uitdrukking/statement gemaakt kunnen worden. Een voorbeeld; de instantie met id 'CBVS' is een instantie van type 'OverheidsOrganisatie'. Hieruit kan worden afgeleid dat diezelfde instantie ook een instantie is van type 'Organisatie', zie afbeelding 7. Een tweede voorbeeld. Een relatie :ligtIn wordt gedefinieerd als zijnde transitief, zie afbeelding 8.

Dit zijn eerder triviale voorbeelden. RDFS en zeker OWL bieden veel meer modelleringsmogelijkheden om allerlei logische gevolgtrekkingen te kunnen maken. Afbeelding 9 toont de beschrijving van een bepaalde wijn; één maal zonder en één maal met gevolgtrekkingen met behulp van een IDE voor semantische webtoepassingen, Topbraid Composer. De op basis van het model afgeleide statements hebben een blauwe achtergrond.
In het semantisch web wereldje vindt men twee richtingen: de school met de slogan "A little semantics goes a long way" versus de big O aanhangers. De eerste school gaat pragmatisch te werk en modelleert alleen wat nodig is om die inferences te verkrijgen die men nodig heeft om bijvoorbeeld tot een betere data-integratie te komen. De tweede school wil een domein in zijn totaliteit en zo volledig mogelijk modelleren. OWL-DL reasoners worden dan gebruikt voor de volgende functionaliteiten:

Statement	Inference
:maker a owl:FunctionalProperty :X :maker :A :X :maker :B	:A owl:sameAs :B

Afbeelding 10.

- het automatisch classificeren van de verschillende klassen als sub- en superklassen van elkaar (subsumption);
- het vinden van inconsistenties, contradicties, klassen die nooit een instantie kunnen hebben (unsatisfiable classes).

Semantische technologie en data-integratie

Dezelfde entiteit (bijvoorbeeld een wijn) kan verschillende unieke identifiers hebben (conform de Non-Unique Naming Assumption). Met de property 'owl:sameAs' kunnen we expliciet maken dat de resources met verschillende identifiers toch één en dezelfde resource zijn, met als gevolg de merging van de respectievelijke data. Maar dit kan ook impliciet, door een bepaalde property die slechts één waarde per individu kan aannemen als functioneel aan te duiden (bijvoorbeeld bij wijn property 'maker'). Zie afbeelding 10.
Zo biedt RDFS en OWL een hele reeks van modelleermogelijkheden om af te leiden dat instances, maar ook property's en klasses gelijk, equivalent zijn. Zo kunt u gemakkelijk aanduiden dat de property met naam A uit database X eigenlijk hetzelfde betekent als de property met naam B uit database Y. Dit alles met slechts één doel om data, die al gemakkelijk te mergen waren dank zij het graph datamodel, nog verder te integreren.

Wat biedt het semantische web?

RDF is een datamodel dat ons toelaat om data op een decentrale manier te creëren en zeer eenvoudig te mergen. Door het volgen van een aantal zeer eenvoudige regels kunnen die data ook op het web gepubliceerd, doorzocht en als linked data benavigeerd worden. Door het toevoegen van enkele formele logische regels kan men dan bovendien nieuwe data afleiden.
Deze collectie van standaarden en de ondersteunende toolsets krijgen voldoende maturiteit om ook binnen het bedrijf hun plaats in data-integratie scenario's te vinden. Van overheden wordt verwacht dat zij hun data open maken en publiceren (zoals in de UK en US); publiceren als Linked Data lijkt dan een voor de hand liggende keuze. *Web 3.0 komt dichterbij.*

Literatuur

Practical RDF, Shelley Powers, O'Reilly, 2003.
Semantic Web for the Working Ontologist, Dean Allemang & Jim Hendler, Morgan Kaufmann, 2008.
Semantic Web for Dummies, Jeffrey T. Pollock, Wiley, 2009.
Semantic Web Programming, John Hebeler et al., Wiley, 2009.
Programming the Semantic Web, Toby Segaran et al., O'Reilly, 2009.
Foundations of Semantic Web Technologies, Pascal Hitzler et al., 2009.

Paul Hermans (paul@proxml.be) is onafhankelijk developer en consultant op het vlak van XML en RDF/OWL- technologieën.

Noot redactie

Uit dit artikel hebben wij helaas de niet-reproduceerbare illustraties moeten weglaten. Het integrale artikel met alle screenshots en actieve links kunt u op www.dbm.nl/Speciaal/Extra-materiaal vinden.