

Paul Hermans

Web 3.0, Linked Data en semantische technologie voor data-integratie en mashups

Inleiding	1
Bouwstenen van het semantisch web	1
RDF (Resource Description Framework): het datamodel voor het web.	1
<i>Voordeel</i>	1
<i>Nog niet populair?</i>	1
<i>Hoe maak ik RDF?</i>	1
SPARQL, de bijhorende query taal.	1
Linked Data	1
Principes	1
<i>Hoe maak je URL 's die een ding identificeren dereferenceable?</i>	1
<i>Het linken naar andere URL 's.</i>	1
<i>Linked data browsers</i>	1
<i>Linked data search engines</i>	1
Hoe verhouden Linked Data zich tot Web 2.0 API's?	1
Voorbeeld van een linked data applicatie: BBC Music	1
Waar zit de semantiek in de Linked Data aanpak?	1
Meer semantiek: ontologieën	1
Semantiek gedefinieerd in termen van logische conclusies	1
Het gebruik van semantische technologieën voor data integratie	1
Samenvatting	1
Wat biedt het semantische web ons?	1
Gebruik in bedrijf en overheid	1
Conclusie	1
Resources	1

Specificaties	2
Tools	2
<i>Linked Data browsers</i>	2
<i>Linked Data search engines</i>	2
<i>SPARQL Endpoints</i>	2
<i>Triple Stores</i>	2
<i>Reasoners</i>	2
<i>IDE's</i>	2
Boeken	2
Credits	2

Inleiding

Er bestaat immens veel verwarring over wat het semantisch web is en wat het inhoudt. Dit artikel zet één en ander op een rij en probeert ook telkens 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 waar weinig semantiek komt bij kijken, te eindigen bij de wereld van description logics en het gebruik van full-blown ontologies.

Bouwstenen van het semantisch web

RDF (Resource Description Framework): 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 eender welk onderwerp. 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 éénder welk moment kan nieuwe informatie/nieuwe data opduiken. Dit is de **Open World Assumption**. Dit uitgangspunt heeft allerlei onverwachte consequenties als we gaan modeleren en reasonen.

Omdat iedereen uitspraken kan doen over eender welk onderwerp betekent dit ook dat elkeen 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.

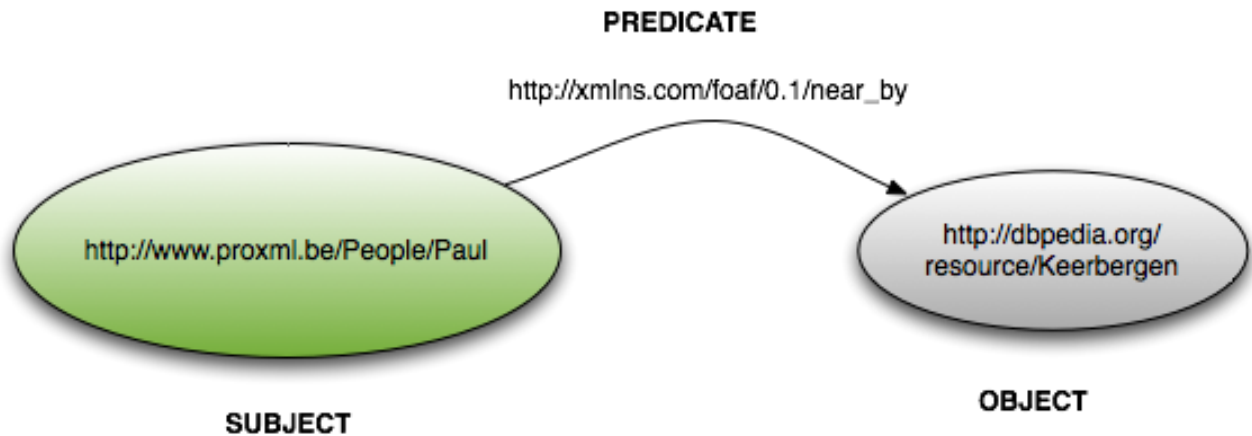
RDF is bijgevolg schemaloos; en iedereen kan relaties leggen tussen paren van resources (of een resource en een atomaire datavalue).

Een voorbeeld van zo'n statement van een relatie tussen 2 resources:

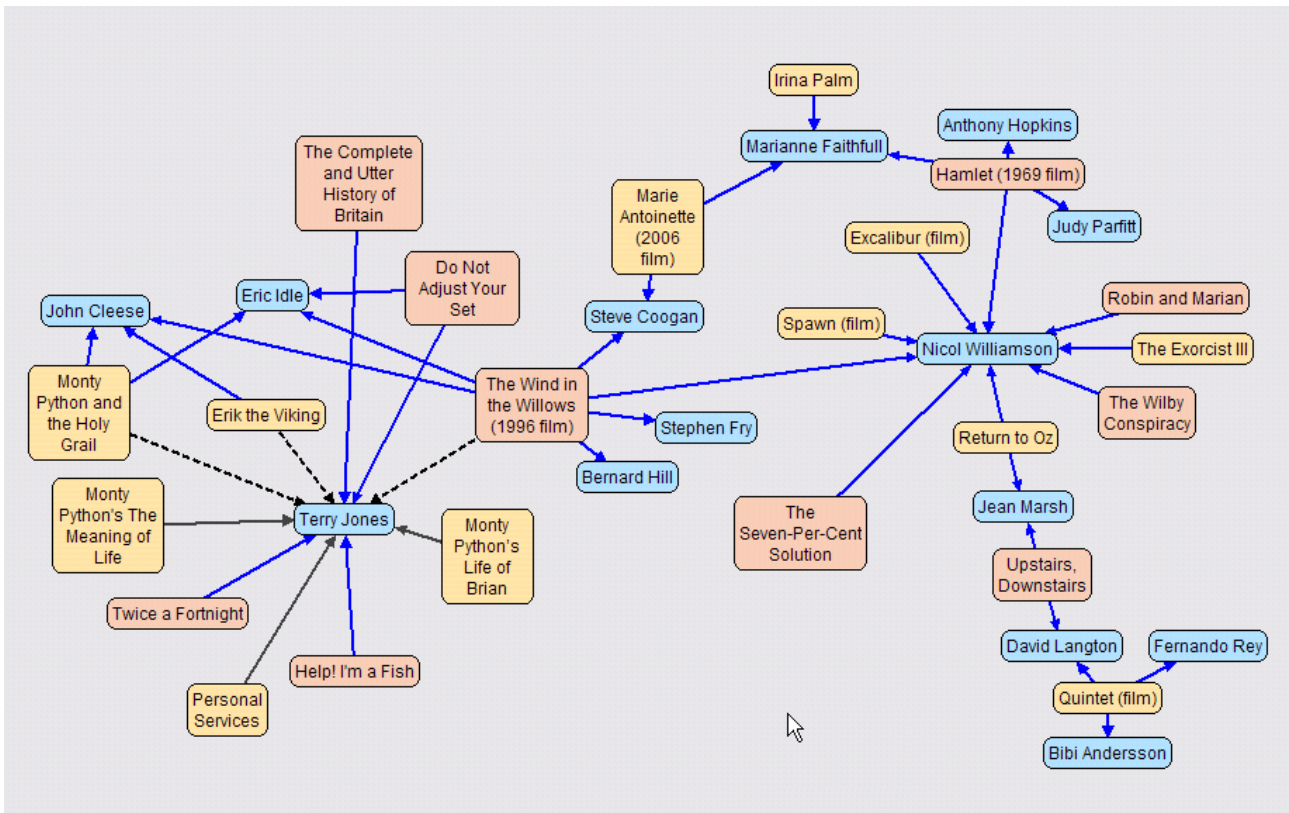
- een persoon met unieke identifier, b.v. <http://www.proxml.be/People/Paul>
- een gemeente met unieke identifier, b.v. <http://dbpedia.org/resource/Keerbergen>

Een benoemde relatie, nl. http://xmlns.com/foaf/0.1/near_by tussen die 2.

Dit noemt men een **triple**. Zo'n triple bestaat uit een subject, een predicate en een object.



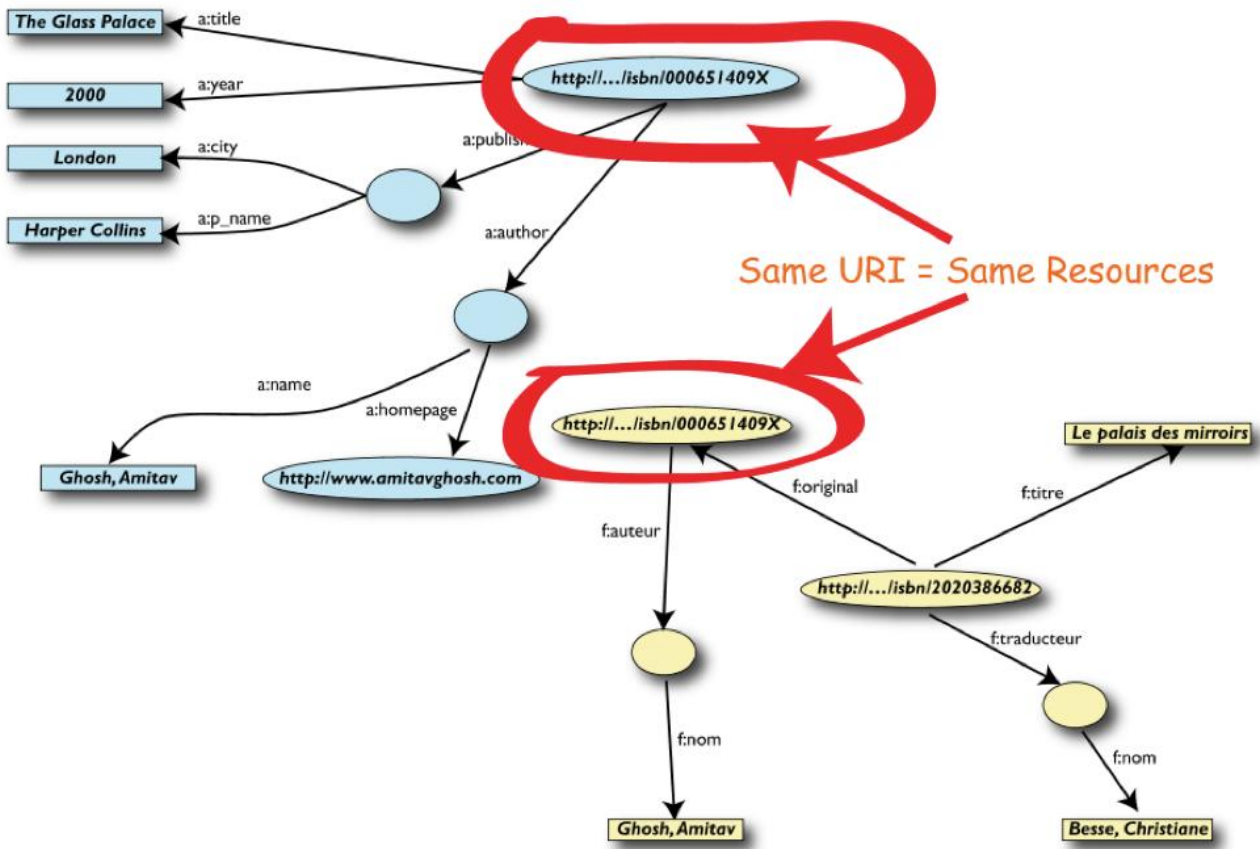
Vele triples samen vormen een **graph**.



Voordeel

Wat is nu het grote voordeel van dit graph data model? 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, nl. het gemak waarmee deze data-integratie toelaat.

Een aandachtige lezer zal opwerpen, wat indien mijn data geen gemeenschappelijke identifiers hebben? Dan moeten we meer truken uit de semantische webdoos halen, maar daarover meer in het deel: ontologisch modeleren.



Nog niet populair?

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, Turtle ... ondertussen de overhand.

U hoeft zich daar niet te veel van aan te trekken want er zijn ondertussen voldoende libraries die al die serialisatieformaten kunnen converteren naar elkaar. Maar, 'to set the record straight': RDF is geen XML.

Hoe maak ik RDF?

Er bestaan heel wat tools en libraries om legacy data en formaten zoals RDBM's tabellen en views, Excel spreadsheets, CSV, ... om te zetten naar RDF.

Hieronder een screendump van de mogelijkheden die één semantische web IDE, Topbraid Composer, biedt.

-  GRDDL Data Source Connection
-  OWL File from Spreadsheet (Text Files)
-  OWL Files from UML File
-  OWL Files from XML Schemas
-  OWL Spreadsheet Ontology Instance File from Excel
-  RDF/OWL File from the Web
-  RDF/OWL from Email Connection
-  RDF/OWL from HTML with Calais Text Analysis
-  RDF/OWL Library from the Web
-  RDF/OWL View on Relational Database via D2RQ
-  RDFa Data Source Connection
-  RSS/Atom Feeds Source Connection
-  SPARQL Endpoint Connection

Dus zo moeilijk is het niet om RDF aan te maken.

SPARQL, de bijhorende query taal.

Voor het queryen van 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 queries en het opvragen van resultaten over HTTP.

Een voorbeeld van een SPARQL query:

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

person	collegeOfSpouse
◆ EdwinSchlossberg	◆ Columbia
◆ AristotleOnassis	◆ GeorgeWashingtonU
◆ JohnKennedy	◆ GeorgeWashingtonU
◆ ArnoldSchwarzenegger	◆ GeorgetownUniversity
◆ EdwinSchlossberg	◆ Harvard

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

Principes

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 wou men ook voor data. Het hebben van RDF alleen leidde niet echt tot het gebruik ervan op het web. Daarvoor ontbraken nog een aantal stukjes van de puzzle.

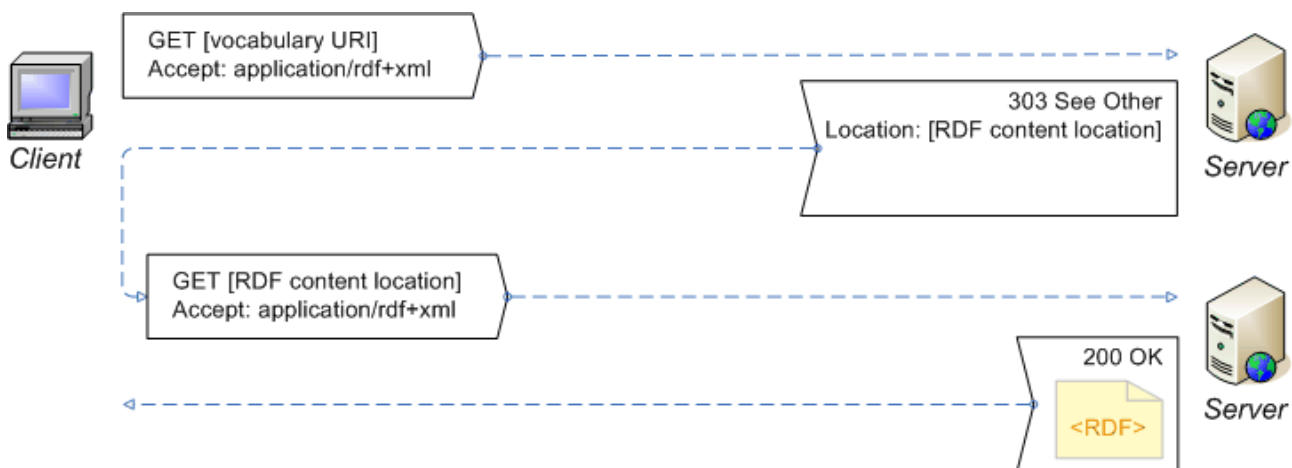
Daarom stelde Tim Berners-Lee in 2006 volgende richtlijnen voor:

- Gebruik URI's als identifier van entiteiten/dingen zoals personen, organisaties, boeken, genen, ...
- Gebruik HTTP URI's zodat deze kunnen worden opgevraagd, m.a.w. 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
- 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 een ding 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 b.v. <http://www.proxml.be/People/Paul.rdf>.



Het linken naar andere URL's.

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.

In het voorbeeld heb ik gezocht op de string “John Zorn”. De componist en saxofonist met naam “John Zorn” heeft o.a. (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 schaalde deze aanpak niet echt; vandaar dat er automatische link generatie hulpmiddelen zoals Silk (<http://www4.wiwiwiss.fu-berlin.de/bizer/silk/>) komen.

Linked data browsers

Linked data kunnen ook gebrowseed 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.

Linked data search engines

Er zijn een aantal search engines ontwikkeld die de linked data cloud crawlen door RDF links te volgen: SWSE, Falcons, Swoogle, Sindice, Watson.

Hoe verhouden Linked Data zich tot 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 resultaat formaat is XML(SOAP of andere), JSON, ... 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, derefereceable over HTTP en RDF als formaat. “Any data, one API.” De scope van de data is bovendien ongebonden: het gaat om één globale dataspace.

Voorbeeld van een linked data applicatie: BBC Music

De Music site van de BBC (<http://www.bbc.co.uk/music/>) is een site gebouwd gebruik makend van allerlei eigen BBC data gekoppeld aan twee publieke data sets: Musicbrainz en DBPedia.

John Zorn

Born 02 September 1952.
John Zorn performs as Rav Tzitzit.

PLAYED MOST ON **BBC**
RADIO



David Redfern/Redferns

Biography

John Zorn (born September 2, 1953 in Queens, New York City) is an American avant-garde composer, arranger, record producer, saxophonist and multi-instrumentalist. Zorn's recorded output is prolific with hundreds of album credits as a performer, composer, or producer. His work has touched on a wide range of musical genres, often within a single composition, but he is best-known for his avant-garde, jazz, improvised and contemporary classical music. Zorn has led the punk jazz band Naked City, the klezmer-influenced quartet Masada and composed the associated 'Masada Songbooks', written concert music for classical ensembles, and produced music for film and documentary. Zorn has stated that "I've got an incredibly short attention span. My music is jam-packed with information that is changing very fast... All the various styles are organically connected to one another. I'm an additive person - the entire storehouse of my knowledge informs everything I do. People are so obsessed with the surface that they can't see the connections, but they are there."

Played By

Since December 2008

Stuart Maconie's Freak Zone

BBC 6 Music

Stuart plays strange and beautiful tracks from a host of freaky artists

Information displayed about artists played on BBC programmes is incomplete at present. [Find out more about this artist play count information.](#)

Played On

Since December 2008

BBC 6 Music

Information displayed about artists played on BBC radio networks is incomplete at present. [Find out more about this artist play count information.](#)

Links

Wikipedia article on [John Zorn](#)

IMDb at imdb.com/name/nm0957958

MySpace at myspace.com/johnzorn

Last.fm page on [John Zorn](#)

MusicBrainz entry on [John Zorn](#)

Waar zit de semantiek in de Linked Data aanpak?

Er zit enkel semantiek in het property deel van onze data triples. Properties hebben meer dan een label; zij hebben een unieke identifier. In het semantische web datamodel zijn properties first class citizens. Properties staan op zichzelf. Daarin verschilt dit van het traditionele OO of RDBMS denken waar properties 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) vocabulary dat properties definieert om personen, de linken tussen hen en de dingen die zij maken en doen te beschrijven. Dit vocabulary is goed gedocumenteerd zodat voor iedereen duidelijk is wat de preciese semantiek van zo'n property is.

Meer semantiek: ontologieën

Semantiek gedefinieerd in termen van logische conclusies

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 leggen we deze kennis zo formeel mogelijk vast zodoende dat ook machines hiermee kunnen werken.

Voor gebruik op het web zijn hiervoor een aantal ontologytalen ontwikkeld: RDFS en OWL in zijn verschillende versies, subtalen en profielen. Deze varianten zijn er omdat expressiviteit een kost heeft. Hoe expressiever (hoe meer men formeel kan vastleggen) de ontologytaal, hoe moeilijker het is om software te maken die binnen een redelijke tijd, juiste en volledige afleidingen kan maken. De OWL-FULL species die het meest expressief is, is zelfs "undecidable".

Enkele voorbeelden van ontologische statements.

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

Deze RDFS/OWL statements zijn zoals u kan zien ook RDF **triples**, wat betekent dat data statements en model statements hetzelfde datamodel volgen en dus geïntegreerd kunnen gebruikt worden met dezelfde tools, query talen, ... Een immens voordeel.

Wat is nu precies de semantiek van deze ontologische statements? De semantiek is geformaliseerd door te specificeren welke "inferences", logische gevolgtrekkingen er op basis van een 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".

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

Een tweede voorbeeld. Een relatie :ligtIn wordt gedefinieerd als zijnde transitief.

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

Dit zijn eerder triviale voorbeelden. RDFS en zeker OWL bieden veel meer modeleringsmogelijkheden om allerlei logische gevolgtrekkingen te kunnen maken. Ik toon u even de beschrijving van een bepaalde wijn één maal zonder en één maal met gevolgtrekkingen m.b.v. een IDE voor semantische web toepassingen, Topbraid Composer. De op basis van het model afgeleide statements hebben een blauwe achtergrond.

Resource Form

Name: OK

Annotations	Other Properties
Incoming References	rdf:type <input type="text" value="vin:Merlot"/>
	vin:hasBody <input type="text" value="vin:Light"/>
	vin:hasFlavor <input type="text" value="vin:Moderate"/>
	vin:hasMaker <input type="text" value="vin:Longridge"/>
	vin:hasSugar <input type="text" value="vin:Dry"/>
	vin:locatedIn <input type="text" value="vin:NewZealandRegion"/>

zonder inferences

Resource Form

Name: OK

Annotations	Other Properties
Incoming References	rdf:type <input type="text" value="food:Wine"/> <input type="text" value="vin:DryRedWine"/> <input type="text" value="vin:DryWine"/> <input type="text" value="vin:Merlot"/> <input type="text" value="vin:RedTableWine"/> <input type="text" value="vin:RedWine"/> <input type="text" value="vin:TableWine"/> <input type="text" value="vin:Wine"/>
← vin:madeIntoWine <input type="text" value="vin:MerlotGrape"/>	food:madeFromFruit <input type="text" value="vin:MerlotGrape"/>
← vin:producesWine <input type="text" value="vin:Longridge"/>	vin:hasBody <input type="text" value="food:Light"/> <input type="text" value="vin:Light"/>
	vin:hasColor <input type="text" value="food:Red"/>

met inferences

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 modeleert alleen wat nodig is om die inferences te verkrijgen die men nodig heeft om b.v. tot een betere dataintegratie te komen.

De tweede school wil een domein in zijn totaliteit en zo volledig mogelijk modeleren. OWL-DL reasoners worden dan gebruikt voor de volgende functionaliteiten:

- het automatisch classificeren van de verschillende klassen als sub- en superklassen van mekaar (subsumption)

- het vinden van inconsistenties, contradicties, klassen die nooit een instantie kunnen hebben (unsatisfiable classes).

Het gebruik van semantische technologieën voor data integratie

Dezelfde entiteit (b.v. een wijn) kan verschillende unieke identifiers hebben (cf. de Non-Unique Naming Assumption).

Met de property ‘owl:sameAs’ kunnen we expliciet maken dat de resources met verschillende identifier toch één en dezelfde resource is, met als gevolg de merging van de respectieve data.

Maar dit kan ook impliciet, door een bepaalde property die slechts één waarde per individu kan aannemen als functioneel aan te duiden (b.v. bij wijn property ‘maker’)

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

Zo biedt RDFS en OWL een hele reeks van modelemogelijkheden om af te leiden dat instances, maar ook properties en klassen gelijk, equivalent zijn.

Zo kan 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.

Samenvatting

Wat biedt het semantische web ons?

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.

Gebruik in bedrijf en overheid

Deze collectie van standaarden en de ondersteunende toolsets krijgen voldoende maturiteit om ook binnen het bedrijf hun plaats in dataintegratie scenario's te vinden.

Van overheden wordt verwacht dat zij hun data open maken en publiceren (cf. UK, US, ...); als Linked Data lijkt dan een voor de hand liggende keuze.

Conclusie

Web 3.0 komt dichterbij.

Resources

Specificaties

- RDF <http://www.w3.org/TR/rdf-syntax-grammar/>
- SPARQL <http://www.w3.org/TR/rdf-sparql-query/> en <http://www.w3.org/TR/rdf-sparql-protocol/>
- RDFS <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>
- OWL <http://www.w3.org/TR/owl-features/>

- Linked Data <http://www.w3.org/DesignIssues/LinkedData>

Tools

Linked Data browsers

- Tabulator <http://dig.csail.mit.edu/2005/ajar/ajaw/About.html>
- Disco <http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/>
- ODE <http://esw.w3.org/topic/OpenLinkDataExplorer>
- Zitgist <http://dataviewer.zitgist.com/>
- Marbles <http://marbles.sourceforge.net/>

Linked Data search engines

- Falcons <http://iws.seu.edu.cn/services/falcons/objectsearch/index.jsp>
- Sindice <http://www.sindice.com/>
- Watson <http://watson.kmi.open.ac.uk/WatsonWUI/>
- SWSE <http://www.swse.org/>

SPARQL Endpoints

- <http://demo.openlinksw.com/sparql/>
- <http://www.sparql.org/sparql.html>

Triple Stores

- Sesame <http://www.openrdf.org/>
- Jena <http://jena.sourceforge.net/> (more than a triple store, a framework)
- OpenAnzo <http://www.openanzo.org/>
- Virtuoso <http://virtuoso.openlinksw.com/>
- AllegroGraph <http://www.franz.com/agraph/allegrograph/>
- Oracle http://www.oracle.com/technology/tech/semantic_technologies/pdf/semantic_tech_rdf_wp.pdf

Reasoners

- Pellet <http://clarkparsia.com/pellet/>
- Fact++ <http://owl.man.ac.uk/factplusplus/>
- RacerPro <http://www.racer-systems.com/products/racerpro/index.phtml>

IDE's

- TopBraid Composer http://www.topquadrant.com/products/TB_Composer.html
- Protégé <http://protege.stanford.edu/>

Boeken

- Practical RDF, Shelley Powers, O'Reilly, 2003
- Semantic Web for the Working Ontologist, Dean Allemang & Jim Hendler, Morgan Kaufmann, 2008 (goede inhoud, maar vol met typos, foute tekeningen, ... een schande op het blazoen van Elsevier)
- Semantic Web for Dummies, Jeffrey T. Pollock, Wiley, 2009 (blijft wat te veel hangen in marketingpraat)
- Semantic Web Programming, John Hebler et al., Wiley, 2009 (Java oriented)
- Programming the Semantic Web, Toby Segaran et al., O'Reilly, 2009 (Python oriented)
- Foundations of Semantic Web Technologies, Pascal Hitzler et al., 2009 (logics oriented)

Credits

Paul Hermans is onafhankelijk developer en consultant op het vlak van XML en RDF/OWL technologieën en evangeliseert het gebruik van open en linked data.

Contactinfo: ProXML bvba, XML and OWL/RDF services

(w) www.proxml.be
(b) [experiences and opinions](#)
(e) paul@proxml.be
(t) +32 15 23 00 76
(m) +32 473 66 03 20
Narcisweg 17
3140 Keerbergen
België