

OWL - Web Ontology Language

Hansi Keijonen

Seminaariraportti
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 28. helmikuuta 2013

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Hansi Keijonen			
Työn nimi — Arbetets titel — Title			
OWL - Web Ontology Language			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Seminaariraportti	28. helmikuuta 2013	8	
Tiivistelmä — Referat — Abstract			
Tiivistelmä.			
Avainsanat — Nyckelord — Keywords			
avainsana 1, avainsana 2, avainsana 3			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

Sisältö

1	Semanttinen web	3
2	Teknologiat ja kielet merkitysten ilmaisuun semanttisessa webissä	3
2.1	URI ja XML	3
2.2	Tiedon esittäminen RDF-triploilla	4
2.3	Alkeelliset ontologiat RDF Schemalla	5
2.3.1	Resurssit voidaan ryhmitellä luokiksi, rdfs:Class . . .	5
2.3.2	Ominaisuuksien ylikuokka, rdf:Property	5
2.3.3	Luokan jäsenet ovat sen ilmentymiä, rdf:type	5
2.3.4	Luokalla on aliluokkia, rdf:subClassOf	5
2.3.5	Ominaisuuksilla on aliominaisuuksia, rdf:subPropertyOf	5
2.3.6	Ominaisuuden sovellusaluetta ja sallittuja arvoja voi rajoittaa, rdfs:range ja rdfs:domain	5
2.4	Kehittyneitä ontologioita OWL:llä	5
3	OWL Web Ontology Language	6
3.1	RDF:n ja RDF Scheman rajoitukset	6
3.2	OWL:n kolme alikieltä	7
3.3	OWL-ontologian rakenne	7
3.3.1	Nimiavaruudet	7
4	OWL2	8
	Lähteet	8

1 Semanttinen web

Suurin osa tämän päivän webin sisällöstä on tarkoitettu ihmisten luettavaksi sekä tulkittavaksi. Kone pystyy tulkitsemaan esim. html-tiedoston ja esittämään dokumentin siinä määritellyllä tavalla. Ongelma on, että kone ei *ymmärrä* dokumentin sisällön merkitystä, semantiikkaa [2]. Se, että kone ei ymmärrä dokumenttien semanttisia merkityksiä rajoittaa esimerkiksi haut internetissä olevista dokumenteista yksinkertaiseksi hakusanojen etsimiseksi. Sen sijaan jos hakukoneet ymmärtäisivät asioiden merkityksen ja niiden välillä vallitsevat yhteydet, olisi hakukoneiden hakutulokset tarkempia ja sisältäisivät mahdollisesti laajennettuja hakuja alkuperäisen asian ympäriltä [2]. On siis tarve olla menetelmä käsitteiden luomiseen, käsitteiden ominaisuuksien kuvaamiseen sekä käsitteiden välisten suhteiden kuvaamiseen [2]. Tim Berners-Lee, James Hendler ja Ora Lassila toteavat artikkelissaan "Semantic web", että "semanttinen web ei ole erillinen web vaan laajennos tämänhetkiseen webiin, jossa informaatiolle on annettu hyvin muotoiltu merkitys mahdollistaen koneiden ja ihmisten paremman yhteistyön." web of documents -> web of data dataa voi parsia manuaalisesti tai koneellisesti

2 Teknologiat ja kielet merkitysten ilmaisuun semanttisessa webissä

W3c tarjoaa semanttisen webin toteuttamiseen standardit teknologioista ja kielistä. Kuvassa n on semanttisen webin teknologiapino sekä ajatuskonseptteja semanttisen webin toteuttamiseen. Osa teknologioista on todellisuutta ja käytössä, osa vasta ideatasolla. Jokainen kerros käyttää alemman kerroksen palveluita. Seuraavissa kappaleissa käyn läpi kaavion teknologioita ja kieliä alhaalta ylöspäin kohti OWL:ää. Jokaisesta seuraavissa kappaleissa esitetystä teknologiasta käsitellään tarkemmin ne konstruktiot, jotka ovat olennaisia ja käytössä myös OWL:ssä.

2.1 URI ja XML

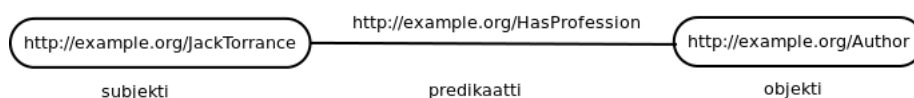
Semanttisessa webissä luokkia, ilmentymiä, ominaisuuksia ja ominaisuuksien arvoja kutsutaan resursseiksi. Jotta sekaannusta jo määritettyjen resurssien sekä uusien määritysten kanssa ei syntyisi, identifoidaan kaikki resurssit (pl. ominaisuuksien literaaliarvot) yksilöllisesti URI(Unified Resource Locator):lla. URI:n avulla voidaan viitata mihin tahansa määritettyyn resurssiin. Useinmiten URI:n toimii perinteinen URL(Unified Resource Locator)-osoite [2]. IRI (Internationalized Resource Identifier) on ainoastaan merkistölajajennos URI:in.

Semanttisen webin datan kuvaukset toteutetaan useimmiten XML-tiedostoina. XML-kieltä voidaan käyttää monimutkaisen rakenteisen tiedon esittämiseen

ja tarjoaa näin standardoidun mallin tiedon vaihtamiseen prosessoijien välillä. Tärkeä XML:n ominaisuus on nimiavaruudet, jotka mahdollistavat resurssien identifiointin URI:en avulla. Tästä kuitenkin enemmän seuraavissa kappaleissa.

2.2 Tiedon esittäminen RDF-triploilla

Resource Description Framework RDF on kieli webissä olevien resurssien kuvaamiseen. RDF perustuu asioiden identifiointiin URI:lla ja näiden asioiden kuvaamiseen ominaisuuksilla ja ominaisuuksien arvoilla. Tämä mahdollistaa yksinkertaisten lausumien esittämisen verkkoina, joissa resurssit ja ominaisuudet ovat soluja ja ominaisuudet verkon kaaria [4]. Kuvassa 1 on havainnollistettu asiaa.



Kuva 1: RDF-tripla joka kuvaa yksinkertaisen lausuman. Jokainen triplan solmu ja kaari on identifioitu URI:lla.

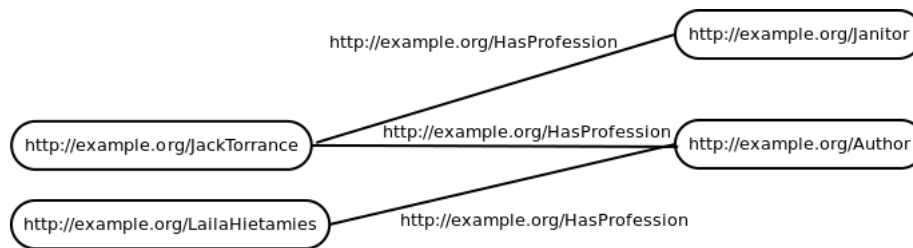
Kuvassa 1 on havainnollistettu yksinkertainen *RDF-tripla*. Triplan subjekti, predikaatti ja objekti kertovat, että "Jack Torrance on ammatiltaan kirjailija". Jokainen solu ja kaari on esimerkissä identifioitu URI:lla. Objekti voi olla myös literaali, jolloin sitä ei identifioida erikseen, mutta formaatti määritellään *****NIIN VITTU MILLÄ?*****

Yleisin tapa esittää tripla on XML-notaatio. Myös muut tavat ovat mahdollisia, kuten esimerkiksi *JSON* ja *turtle*. Alla on esitetty kuvassa 1 triplan XML-notaatio:

```
<rdf:Description rdf:about="http://example.org/JackTorrance">
  <rel:HasProfession rdf:resource="http://example.org/Author"/>
</rdf:Description>
```

Esimerkistä näkee selvästi, kuinka subjekti, predikaatti ja objekti ovat identifioitu URI:lla. Jack Torrance on toiselta ammatiltaan talonmies ja myös Laila Hietamies on ammatiltaan kirjailija. Kun myös nämä triplat kuvataan samaan kaavioon, alkaa pieni mutta informatiivinen verkkokuva ja selitys triploista ja URI:sta? Kyllä, tähän se sopisi kokoavana elementtinä. o syntyä, kuva 2. Tästä verkosta voisimme tehdä hakuja, kuten "ketkä ovat kirjailijoita?".

RDF tarjoaa siis vain melko alkeellisen tavan esittää lausumia, jotka muodostavat haut mahdollistavan verkon. RDF toteuttaa myös joukon muita ominaisuuksia, kuten *säiliöitä* (container) tiedon säilömiseen sekä *kokoomatietorakenteita* (collections) asioiden listaamiseen [4]. Näitä käsitellään myöhemmin niiltä osin kuin ne ovat relevantteja OWL-ontologioiden muodostamisessa.



Kuva 2: RDF-triplat muodostavat verkon.

2.3 Alkeelliset ontologiat RDF Schemalla

Semantiikkaa voidaan webissä ilmaista ontologioilla. Tietojenkäsittelytieteessä ontologialla tarkoitetaan dokumenttia, jossa kerrrotaan asioiden välisistä yhteyksistä [2]. ***vajaa läppä!!*****

RDF:llä ilmaistut ominaisuudet voidaan ajatella resurssien attribuuteiksi samassa mielessä kuin perinteiset attribuutti-arvo -parit [3]. Vaikka RDF:llä voidaan kuvata resursseja, se ei tarjoa keinoja kuvata ominaisuuksia tai ominaisuuksien välisiä suhteita. Tämä on mahdollista RDF:n *sanastonkuvauslaajennoksella* RDF Schemalla. RDF Schemalla on mahdollista määritellä luokkia ja ominaisuuksia joita voidaan käyttää luokkien, ominaisuuksien ja resurssien kuvaamiseen [3]. RDF Schema on RDF:n semanttinen laajennos. Seuraavassa esitellään RDF Scheman peruskonstruktiot, koska myös OWL käyttää niitä lähes sellaisenaan.

2.3.1 Resurssit voidaan ryhmitellä luokiksi, `rdfs:Class`

2.3.2 Ominaisuuksien yliluokka, `rdfs:Property`

2.3.3 Luokan jäsenet ovat sen ilmentymiä, `rdfs:type`

2.3.4 Luokalla on aliluokkia, `rdfs:subClassOf`

2.3.5 Ominaisuuksilla on aliominaisuuksia, `rdfs:subPropertyOf`

2.3.6 Ominaisuuden sovellusaluetta ja sallittuja arvoja voi rajoittaa, `rdfs:range` ja `rdfs:domain`

2.4 Kehittyneitä ontologioita OWL:llä

Yleinen tapa ilmaista ontologioita on OWL (Web Ontology Language)-ontologiat. OWL-ontologioissa määritellään sovellusalueen luokkia, ominaisuuksia, luokkien ilmentymiä eli instansseja sekä instanssien välisiä suhteita [6]. Ontologoiden pohjalta on mahdollista päätellä implisiittisiä asioiden ilmentymien välisiä suhteita. Eräs tapa hahmottaa RDF-triplojen ja OWL-ontologioiden välinen ero on verrata niitä perinteiseen relaatiotietokantaan. RDF-triplat on tapa tallettaa tietoa samalla tavalla kuin relaatiotietokannan riveillä tal-

lennetaan rakenteista tietoa. Jokaista riviä yksilöi yksilöivä avain kun taas RDF-triploissa yksilöinnin takee URI(Uniform Resource Identifier). Vastaa-
vasti asioiden välisiä suhteita kuvataan OWL-ontologioilla samalla tavalla
kuin relaatiotietokannan tauluissa suhteita ilmaistaan taulurakenteilla ja
viitteillä tauluista toiseen. RDF on kuin kieli jolla ilmaistaan asioita kun
taas OWL on sanasto, jonka avulla merkitykset ymmärretään.

kuva ja selitys triploista ja URI:sta? Kyllä, tähän se sopisi kokoavana
elementtinä.

3 OWL Web Ontology Language

mitä owl tarjoaa? syntaksi, semantiikka, tuki päättelylle, ilmaisuvoimaa,
käytettävyyttä Description Logics

3.1 RDF:n ja RDF Scheman rajoitukset

RDF/RDFS:ssä on toteutettu luokka (class) ja sen aliluokka (subclass), suhde
(property) ja siitä periytynyt alisuhde (subproperty), suhteen sovellusalue ja
arvorajoitukset sekä luokkien ilmentymien väliset suhteet [1]. Mutta useita
tarpeellisia ominaisuuksia puuttuu, jotka on toteutettu OWL:ssä, esimerkiksi
[1]:

- *Rajoitukset suhteen kohteessa:* (WTF?) RDFS:ssä voidaan määritellä rajoituksia, mitä arvoja tai muita luokkia suhteen objektiksi voidaan asettaa. Kuitenkaan ei ole mahdollista määrätä rajoitteita sen suhteen, onko jotkut rajoitteet voimassa vain rajatulle joukolle luokkia.
- *Luokkien poissulkevuus:* joissain tapauksissa on hyvä pystyä sanomaan, että joku luokka on poissulkeva toisen luokan suhteen. Voidaan sanoa esimerkiksi että luokka 'mies' on poissulkeva suhteessa luokkaan 'nainen'
- *Luokkien joukko-operaatiot:* toisinaan on tarve muodostaa uusia luokkia olemassaolevien joukkojen perusteella. Esimerkiksi luokka 'henkilö' voitaisiin määritellä luokkien 'mies' ja 'nainen' yhdisteeksi kuitenkin niin, että henkilö voi olla vain joko mies tai nainen (poissulkevuus).
- *Suhteiden kardinaliteetit:* joihinkin suhteisiin on hyvä määrittää rajoituksia sen suhteen kuinka monta arvoa arvoksi voi antaa. Esimerkiksi ihmisellä voi olla ainoastaan kaksi vanhempaa.
- *Suhteiden erikoisominaisuudet:* suhteet voidaan määritellä *transitiiviksi*, *uniikeiksi* jonkun toisen suhteen *inversioksi*.

Tarvitaan siis kieli, joka on rikkaampi kuin RDF Schema [1]. OWL:n suunnittelussa on jouduttu tasapainoilemaan kielen ilmaisuvoiman ja sen sovel-
tuvuudesta koneelliseen päättelyyn. Mitä rikkaampi ja ilmaisuvoimaisempi

kieli on, sitä hankalampi ja tehottomampi sen pohjalta on päätellä asioita. Pahimmassa tapauksessa päättely muodostuu mahdottomaksi tehtäväksi [1].

Ideaalisesti OWL on RDF:n ja RDF Scheman laajennos [1]. OWL käyttää RDF:n luokkia ja suhteita lisäten niihin omia laajennoksiaan kuten yllä on selitetty. RDF Schemassa on joitain hyvin vahvoja primitiivejä, kuten `rdf:Class` (kaikkien luokkien ylliluokka) sekä `rdf:Property` (kaikkien suhteiden ylliluokka). Näiden primitiivien ilmaisuvoima yhdistettynä yllä luetelluilla laajennoksilla on ristiriidassa sen tavoitteen kanssa, että ontologiat olisivat koneellisesti pääteltävissä. Tämä tasapainotila mielessäpitäen on määritelty kolme OWL:n alikieltä sen perusteella, painotetaanko ilmaisuvoimaa vai koneellista päättelyä [1].

3.2 OWL:n kolme alikieltä

W3C:n Web Ontology Working Group on määritellyt OWL:lle kolme alikieltä, joiden on tarkoitus toteuttaa eri aspektit (ilmaisuvoima, koneellinen päättely), joita ontologioiden kuvaamiskieleltä vaaditaan [5]:

- *OWL Full*
- *OWL DL*
- *OWL Lite*

3.3 OWL-ontologian rakenne

3.3.1 Nimiavaruudet

XML-dokumentissa tulee määritellä nimiavaruudet (namespace). Nimiavaruuksien avulla voidaan ratkaista mm. samannimisten elementtien aiheuttamia tulkintaongelmia sekä kertoa lukijalle (koneelle tai ihmiselle) konteksti, jonka mukaan elementtien kahvoja tulee tulkita. OWL-ontologiassa nimiavaruudet määritellään `rdf:RDF` -kahvojen sisään. Alla olevassa esimerkissä on eräs mahdollinen nimiavaruusmäärittely [2].

```
<rdf:RDF
  xmlns = "http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
  xmlns:vin = "http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
  xml:base = "http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
  xmlns:food = "http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#"
  xmlns:owl = "http://www.w3.org/2002/07/owl#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema#">
```


Esimerkin nimiavaruusmäärittelyssä on määritelty oletusnimiavaruus niille kahvoille, joissa ei käyteä mitään etuliitettä sekä nimiavaruudet niille kahvoille, joissa käytetään etuliitteitä vin: tai food:. Nimiavaruudet on määritelty myös owl:-, rdf:- ja rdfs:-etuliitteille kertomaan, että näillä etuliitteillä varustetut kahvat edustavat OWL:n, RDF:n ja RDFS:n termistöä. OWL-ontologia on riippuvainen myös XMLSchema-datatyypeistä, joten myös niiden nimiavaruus tulee määrittää.

4 OWL2

Lähteet

- [1] Antoniou, Grigoris ja Harmelen, Frank Van: *Web Ontology Language: OWL*. Handbook on ontologies, 2(September):91–110, 2009. <http://www.springerlink.com/index/10.1007/978-3-540-92673-3>.
- [2] Berners-Lee, Tim, Hendler, James ja Lassila, Ora: *The Semantic Web*. Scientific American, 284(5):34–43, 2001. <http://www.nature.com/doifinder/10.1038/scientificamerican0501-34>.
- [3] Brickley, Dan ja Guha, R V: *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation, 2009(10 February 2004), 2004. <http://www.w3.org/TR/rdf-schema/>.
- [4] Manola, Frank ja Miller, Eric: *RDF Primer*. W3C Recommendation, 10(February):1–107, 2004. <http://www.w3.org/TR/rdf-primer/>.
- [5] McGuinness, Deborah L ja Van Harmelen, Frank: *OWL Web Ontology Language Overview*. W3C recommendation, 10(February):1–22, 2004. <http://www.w3.org/TR/owl-features/>.
- [6] Smith, Michael K, Welty, Chris ja McGuinness, Deborah L: *OWL Web Ontology Language Guide*. W3C Recommendation, 10(February):1–46, 2004. <http://www.w3.org/TR/owl-guide/>.