

# OWL - Web Ontology Language

Hansi Keijonen

20. helmikuuta 2013

## **Tiivistelmä**

huhhahhei ja sahtia kaivoon

# 1 Semanttinen web

Suurin osa tämän päivän webin sisällöstä on tarkoitettu ihmisten luettavaksi sekä tulkittavaksi. Kone pystyy tulkitsemaan esim. html-tiedoston ja esittämään tietoja tiedostossa 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 tietoja haetun asian ympäriltä [2].

Tapa, jolla ilmaistaan asioiden ominaisuuksia webissä on tripla. Triplassa on subjekti, predikaatti ja objekti, joilla voidaan ilmaista asiayhteyksiä kuten 'hauki(subjekti) on(predikaatti) kala(objekti)'. Triploja voidaan ilmaista XML (Extensive Markup Language)-kieleen perustuvalla RDF(Resource Description Framework)-kielellä. Triploilla ilmaistusta datasta muodostuu varkkotietokanta, josta hakuja tekemällä voidaan hakea tietoa monipuolisilla tavoilla. Tämä verkkotietokanta ei sinällään ole vielä riittävä, koska siinä on ainoastaan talletettuna tietoa asioiden välisistä yhteyksistä, ei merkityksiä. RDF-triploilla ei ole mahdollista tallettaa semantiikoita [2].

Semantiikkaa voidaan webissä ilmaista ontologioilla. Tietojenkäsittelytieteessä ontologialla tarkoitetaan dokumenttia, jossa kerrrotaan asioiden välisistä yhteyksistä [2]. Yleinen tapa ilmaista ontologioita on OWL(Web Ontology Language)-ontologiat. OWL-ontologioissa määritellään sovellusalueen luokkia, ominaisuuksia, luokkien ilmentymiä eli instansseja sekä instanssien välisiä suhteita [4]. 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 tallennetaan rakenteista tietoa. Jokaista riviä yksilöi yksilöivä avain kun taas RDF-triploissa yksilöinnin takee URI(Uniform Resource Identifier). Vastaavasti 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.

teknologiastäkki?

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

## 2 OWL

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

### 2.1 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 *transitiivisiksi*, *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 soveltuvuudesta koneelliseen päättelyyn. Mitä rikkaampi ja ilmaisuvoimaisempi kieli on, sitä hankalampi ja tehottomampi sen pohjalta on päätellä asioita. Pahimmassa tapauksessa päättely muodostuu mahdottomaksi tehtäväksi [1].

Idealisesti 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 ylikuokka) sekä `rdf:Property` (kaikkien suhteiden ylikuokka). 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].

## 2.2 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 [3]:

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

## 2.3 OWL-ontologian rakenne

### 2.3.1 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 nimiavaruusmäärittelyissä 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.

## Viitteet

- [1] Grigoris Antoniou and Frank Van Harmelen. Web ontology language: Owl. *Handbook on ontologies*, 2(September):91–110, 2009.
- [2] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.
- [3] Deborah L McGuinness and Frank Van Harmelen. Owl web ontology language overview. *W3C recommendation*, 10(February):1–22, 2004.
- [4] Michael K Smith, Chris Welty, and Deborah L McGuinness. Owl web ontology language guide. *W3C Recommendation*, 10(February):1–46, 2004.