

# **OWL - Web Ontology Language**

Hansi Keijonen

Seminaariraportti  
HELSINGIN YLIOPISTO  
Tietojenkäsittelytieteen laitos

Helsinki, 1. maaliskuuta 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	1. maaliskuuta 2013	10	
Tiivistelmä — Referat — Abstract			
Tiivistelmä.			
Avainsanat — Nyckelord — Keywords			
avainsana 1, avainsana 2, avainsana 3			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

# Sisältö

<b>1</b>	<b>Semanttinen web</b>	<b>3</b>
<b>2</b>	<b>Teknologiat ja kielet jotka mahdollistavat OWL:n</b>	<b>3</b>
2.1	URI ja XML . . . . .	3
2.2	Tiedon esittäminen RDF-triploilla . . . . .	4
2.3	Alkeelliset ontologiat RDF Schemalla . . . . .	5
2.3.1	Luokat ja ilmentymät . . . . .	6
2.3.2	Luokan aliluokka . . . . .	6
2.3.3	Ominaisuudet ja periytetyt ominaisuudet . . . . .	6
2.3.4	Rajoitukset ominaisuuksien sovellusalueissa ja arvoissa	7
2.3.5	RDF Schema esimerkki . . . . .	7
<b>3</b>	<b>Kehittyneitä ontologioita OWL:llä</b>	<b>7</b>
<b>4</b>	<b>OWL Web Ontology Language</b>	<b>8</b>
4.1	RDF:n ja RDF Scheman rajoitukset . . . . .	8
4.2	OWL:n kolme alikieltä . . . . .	9
4.3	OWL-ontologian rakenne . . . . .	9
4.3.1	Nimiavaruudet . . . . .	9
<b>5</b>	<b>OWL2</b>	<b>10</b>
	<b>Lähteet</b>	<b>10</b>

## 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

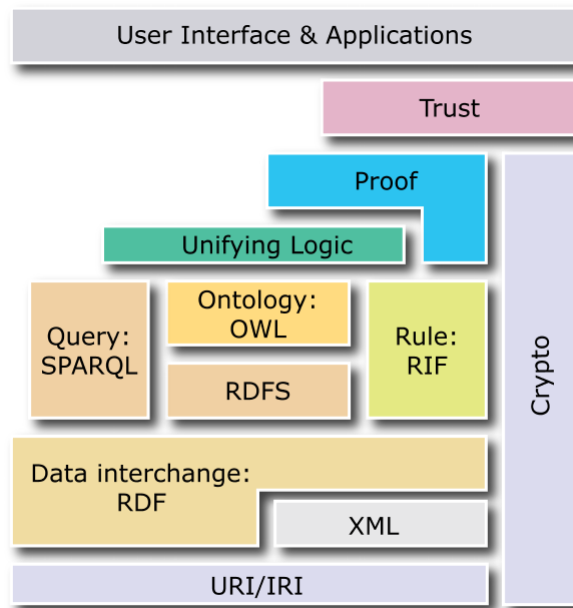
## 2 Teknologiat ja kielet jotka mahdollistavat OWL:n

W3c tarjoaa semanttisen webin toteuttamiseen standardit teknologioista ja kielistä. Kuvassa 1 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.

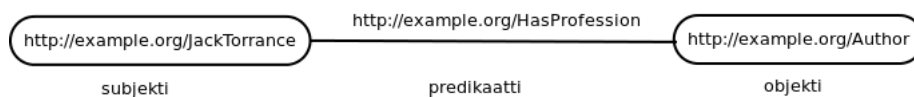


Kuva 1: Semanttisen webin toteutukseen tarvittavat teknologia, kielet ja konseptit.

Tärkeä XML:n ominaisuus on nimiavaruudet, jotka mahdollistavat resurssien identifiointiin 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 2 on havainnollistettu asiaa.



Kuva 2: RDF-tripla joka kuvaa yksinkertaisen lausuman. Jokainen triplan solmu ja kaari on identifioitu URI:lla.

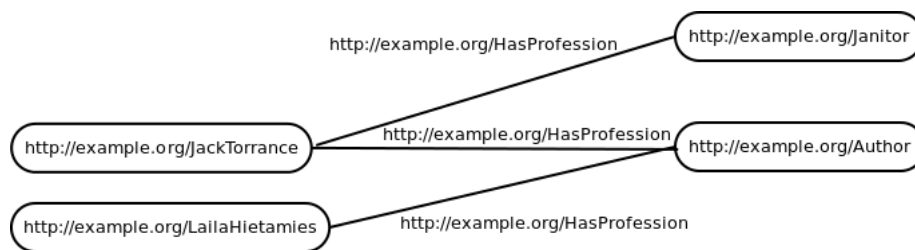
Kuvassa 2 on havainnollistettu yksinkertainen *RDF-tripla*. Triplan subjekti, predikaatti ja objekti kertovat, että "Jack Torrance on ammatiltaan kirjailija". Jokainen solu ja kaari on esimerkissä identifioitu URI:lla. Objekti

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 2 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 verkkkuva ja selitys triploista ja URI:sta? Kyllä, tähän se sopisi kokoavana elementtinä. o syntyä, kuva 3. Tästä verkosta voisimme tehdä hakuja, kuten "ketkä ovat kirjailijoita?".



Kuva 3: RDF-triplat muodostavat verkon.

RDF:n peruskonstruktioita ovat: description, resource, about \*\*\*selitä nämä\*\*\*\*\*

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.

## 2.3 Alkeelliset ontologiat RDF Schemalla

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

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 niitä peruskonstruktioita, joita myös OWL käyttää määrittelyssään lähes sellaisenaan. *RDFS Schema määrittelee myös suurehkon joukon muita konstruktioita, mutta niiden esittely OWL:n selostamisen kannalta on tässä tarpeetonta.*

### 2.3.1 Luokat ja ilmentymät

Resurssit voidaan jaotella luokkiin. Luokkien jäseniä nimitetään luokan ilmentymiksi [3]. Luokat ovat itsessään resursseja. Useimmiten luokat identifioidaan RDF:n URI:lla ja niitä voidaan kuvata RDF:n ominaisuuksilla (property). Luokka ilmaistaan `rdfs:Class` tagilla, `rdf:type` ominaisuudella voidaan ilmaista, että resurssi kuuluu luokkaan [3]. Esimerkiksi kustannusosakeyhtiö "WSOY" on itse luokka ja luokan "Kustantamo" ilmentymä:

```
Esim:WSOY rdf:type rdfs:Class
Esim:WSOY rdf:type esim:Kustantamo
```

### 2.3.2 Luokan aliluokka

`rdfs:subClassOf` ominaisuudella voidaan ilmaista, että jonkin luokan ilmentymät ovat myös jonkin toisen luokan ilmentymiä [3]. Esimerkiksi luokka "Kustantamo" on luokan "Liikkeyritys" aliluokka:

```
esim:Kustantamo rdfs:subClassOf esim:Liikkeyritys
```

### 2.3.3 Ominaisuudet ja periytetyt ominaisuudet

Resursien välisiä suhteita kuvataan ominaisuuksilla. Esimerkiksi "Matilla on veli Teppo". Voidaan ajatella, että resurssien "Matti" ja "Teppo" välinen suhde on ominaisuus "Veljeys". Ominaisuus "Veljeys" voi olla ominaisuuden "Perhesuhde" aliluokka tai paremminkin aliominaisuus. Tällaisia konstruktioita kuvataan RDF Schemassa termillä `subPropertyOf`:

```
esim:Veljeys rdfs:subPropertyOf esim:Perhesuhde
```

`subPropertyOf` ilmaisee määritelmällisesti, että resurssit joiden välistä suhdetta kuvataan jollain suhteella, voidaan kuvata myös tämän suhteen ylisuhteella, josta ko. suhde on periytynyt [3].

### 2.3.4 Rajoitukset ominaisuuksien sovellusalueissa ja arvoissa

RDF Schemassa on mahdollista antaa ominaisuuksille rajoituksia sen suhteen, että minkä luokkien välistä suhdetta ominaisuus kuvaa. Ensinnäkin voidaan rajoittaa suhteen sovellusaluetta (domain). Suhteen "Työskentelee" sovellusalue voidaan rajata koskemaan ainoastaan luokkaa "Yyöntekijä". Samaten suhteen saamat arvot voidaan rajata olemaan ainoastaan luokan "Yritys" tai sen aliluokkien ilmentymiä.

```
esim:Työskentelee rdfs:domain esim:Työntekijä
esim:Työskentelee rdfs:range esim:Yritys
```

### 2.3.5 RDF Schema esimerkki

Myös RDF Scheman varsinainen notaatio on toteutettu XML:llä. Alla on lyhyt esimerkki RDF Schemalla toteutetusta ontologiasta.:

```
<?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="liikeyritys" />

  <rdfs:Class rdf:ID="kustantamo">
    <rdfs:subClassOf rdf:resource="#liikeyritys"/>
  </rdfs:Class>

</rdf:RDF>
```

Esimerkissä on yksinkertainen ontologia, joka kertoo, että "kustantamo" on luokka ja että ko. luokka on myös luokan "liikeyritys" aliluokka. RDF Schema-ontologiat ovat RDF-tiedostoja samoin kuin OWL-ontologiatkin. Tiedoston alussa olevat nimiavaruusmäärittelyt varmistavat, että rdf- ja rdfs-alkuisilla tageilla ympäröidyt elementit todellakin ovat rdf- ja rdfs- elementtejä.

## 3 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 tallennetaan rakenteista tietoa. Jokaista riviä yksilöi yksilöivä avain kun taas



RDF-triploissa yksilöinnin takee URI(Uniform Resource Identifier). Vastavasti 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. valmiit sanastot kuva ja selitys triploista ja URI:sta? Kyllä, tähän se sopisi kokoavana elementtinä.

## 4 OWL Web Ontology Language

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

### 4.1 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 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 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].

## 4.2 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*

## 4.3 OWL-ontologian rakenne

### 4.3.1 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.

## 5 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/>.