

Näkymäpohjainen RDF-haku

Samppa Saarela

Helsinki 26.5.2004

Pro Gradu -tutkielma

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Näkymäpohjainen RDF-haku

Museokokoelmat sisältävät suuret määrät semanttisesti rikasta dataa ja metadataa. Museoesineillä on hyvin dokumentoitu historia ja ne liittyvät monin tavoin nykypäivän ympäristöön, yhteiskuntaan ja muihin museoesineisiin. Tämän kaltaiset suhteet ylittävät museokokoelmien rajat ja liittävät yhteen hyvinkin monenlaisia objekteja, aina historiallisista paikoista käyttöesineisiin ja taideteoksiin. Semantic Web -teknologia mahdollistaa tämänkaltaisen informaation esittämisen ja heterogeenisten tietolähteiden yhdistämisen. Informaatiohaun ja käyttöliittymätutkimuksen parissa kehitetyssä näkymäpohjaisessa hakutavassa yhdistyy hyvä käytettävyys korkeaan löytyvyyteen ja tarkkuuteen. Tässä tutkielmassa osoitetaan kuinka näkymäpohjaista hakua voidaan soveltaa ja edelleen laajentaa Semantic Web -tietämyksen esitystavalla RDF-kielellä.

Näkymäpohjaisessa haussa kysely muodostetaan valitsemalla käsitteitä ihmisen ajattelun kannalta luonnollisista käsittehierarkioista. Esimerkiksi haku ”lintu” palauttaisi myös kohteita, jotka liittyvät kanoihin tai viiriäisiin. Käsittehierarkiat kuvaavat haettavia kohteita useista eri näkökulmista. Näkökulmia voivat olla esimerkiksi *mitä*, *missä* ja *milloin*. Kaikki käsitteet (esimerkiksi aika) eivät kuitenkaan ole luonnollisesti hierarkisia tai niiden hierarkinen esitysmuoto on liian rajoittava. Tällaisia käsitteitä on käytännöllisempää hakea lomakepohjaisesti. Tässä tutkielmassa esitetään näkymäpohjaisen haun laajennus, joka mahdollistaa lomakepohjaisen haun yhdistämisen näkymäpohjaiseen hakuun. Laajassa mittakaavassa vastaavan toiminnallisuuden toteuttaminen perinteisellä tietokantatekniikalla vaatisi näkymäpohjaisen hakumoottorin integroimista kohdetietokantaan.

Esittävä yleinen näkymäpohjainen RDF-hakumoottoriratkaisu perustuu aineistokohtaisten eroavaisuuksien ohittamiseen projektiolla. Tämä projektiio, joka myös esitetään RDF-kielellä, koostuu hierarkia- ja kohdeprojektiosta. Se voidaan laskea joko etukäteen tai ajonaikaisesti. Hakujen näkymäpohjaiset ehdot kohdistuvat tähän projektiioon ja muun tyyppiset ehdot alkuperäiseen dataan tai sen skeemaan. Näkymäpohjaisen haun vaatima projektiio on täten vain yksi metatietokerros alkuperäisen tietämuskannan päällä. Tutkielma vastaa kysymykseen, miten näkymäpohjaista hakuparadigmaa voidaan soveltaa ja edelleen laajentaa RDF-kieleen pohjautuen.

ACM Computing Classification System (CCS): H.2.3 [Languages: Query languages], H.3.3 [Information Search and Retrieval: Search process, Information filtering, Query formulation]

Sisältö

1	Johdanto	1
2	Käsitteistö	4
2.1	Termien ja käsitteiden väliset semanttiset suhteet	5
2.2	Käsitteiden väliset semanttiset suhteet	7
3	Formaali ontologia	9
3.1	Ontologia vai sanasto	9
3.2	Kehykset	11
4	Metatietoarkkitehtuurimalli	13
5	Semantic Web -metatietokielet	15
5.1	Resource Description Framework (RDF)	15
5.1.1	Resurssien kuvaus	16
5.1.2	Reifikaatio	18
5.2	RDF Vocabulary Description Language (RDFS)	20
5.2.1	RDFS ontologiakielenä	21
5.2.2	RDFS metametakielenä	21
5.3	OWL Web Ontology Language	24
5.3.1	Luokkien määrittely	24
5.3.2	Ominaisuuksien määrittely	27
5.3.3	Yksilöt	28
5.3.4	Ontologiat	28
6	Annotaatio	29
6.1	Annotaatio-ontologia	29
6.2	Ontologiapohjainen annotaatio	29
7	Näkymäpohjainen haku	30
7.1	Kohteiden haku	32
7.2	Kategorioiden suodatus	35
7.3	Kohteiden ryhmittely	37
7.4	Näkymäpohjainen sanahaku	39

	iii
7.5 Kohdesivu	41
7.6 Ominaisuusrajoitteet	42
8 Näkymien projisointi	43
8.1 Hierarkiaprojektiot	44
8.2 Kohdeprojektiot	49
9 Ontogator	51
9.1 Syöterajapinta	52
9.2 Kyselyrajapinta	55
9.2.1 Katogoriaehdot	55
9.2.2 Kohde-ehdot	56
9.2.3 Näkymäehdot	57
9.2.4 Optiot	59
9.2.5 Rajoitteet	62
9.3 Vastausrajapinta	63
9.4 Implementaatio	63
10 Ratkaisun arviointia	67
10.1 RDF sovelluskehityksessä	67
10.2 Projektioista	68
10.3 Ontogator-hakumoottorista	70
11 Yhteenveto	72
Lähteet	74
Liitteet	
A Bookmark-ontologia	1
B Ontogator-ontologia	3
C Esimerkki kyselystä Ontogatorille	10
D Esimerkki Ontogatorin vastauksesta	14

1 Johdanto

Museokokoelmat sisältävä suuret määrät semanttisesti rikasta dataa ja metadataa. Museoesineillä on hyvin dokumentoitu historia ja ne liittyvät monin tavoin nykypäivän ympäristöön, yhteiskuntaan ja muihin museoesineisiin. Esimerkiksi tuoli saattaa olla valmistettu pyökistä ja nahasta, olla tunnetun taitelijan suunnittelema, tietyn yrityksen tiettyä aikakautena valmistama ja käytetty määrättyssä linnassa yhdessä monien muiden esineiden kanssa. Tämän kaltaiset suhteet ylittävät museokokoelmien rajat ja liittävät yhteen hyvin monenlaisia objekteja aina historiallisista paikoista käyttöesineisiin ja taideteoksiin.

Semantic Web -teknologia mahdollistaa tämänkaltaisen informaation esittämisen ja heterogeenisten tietolähteiden yhdistämisen [Fen03]. Tässä tutkielmassa keskitytään siihen, miten tätä informaatiota voidaan hyödyntää ja tuoda esille käyttäjälähtöisessä hakupalvelussa.

Hakutulos perinteisellä hakukoneella, esimerkiksi Googlessa, on usein joko tyhjä tai sitten se käsittää useita tuhansia, jopa miljoonia kohteita. Sopivien hakutermin ja sanamuotojen keksiminen ei ole helppoa, eikä käyttöliittymä siinä juurikaan auta. Ongelmia aiheuttavat muun muassa homonymit, eli sanat, jotka samasta kirjoitusasusta huolimatta tarkoittavat eri asioita. Toinen keskeinen ongelma liittyy käsitteiden välisiin semanttisiin suhteisiin kuten hyponymiaan (esimerkiksi kana on lintu) ja meronymiaan (esimerkiksi Orimattila sijaitsee Suomessa). Ihmiselle on ilmeistä, että Orimattilan kotieläintilan kanoja käsittelevät dokumentit ovat relevantteja hakusanojen ”lintu” ja ”Suomi” suhteen. Perinteiset hakukoneet eivät kuitenkaan näitä dokumentteja välttämättä löytäisi.

Perinteisesti hakupalveluja mitataan löytyvyyden (recall) ja tarkkuuden (precision) suhteen [BYRN99]. Siinä missä Semantic Web mahdollistaa paremman löytyvyyden ja tarkkuudenkin [Fen03], eivät nämä yksin riitä. Ihmiset käyttävät hakupalveluja koska heiltä puuttuu toimintaan tai ongelman ratkaisemiseen tarvittavaa tietoa. Käyttäjä saattaa tietää haun kohteesta vähän tai ei mitään. Hänellä ei välttämättä ole myöskään selkeää kuvaa siitä, mitä hän on hakemassa. Heninger ja Belkin korostavat artikkelissaan [HB96], että informaation haku tulisi nähdä kokonaisvaltaisena ongelmanratkaisutapahtumana. Olennaista ei ole ainoastaan tehokas haku vaan käyttäjää tulee myös auttaa itse ongelman määrittelyssä. Informaation haku on iteratiivinen prosessi, jossa käyttäjä muuttaa, tarkentaa tai laventaa hakueta välitulosten perusteella. Tämän kaltainen haku on tyypiltään tutkimushakua (research searches). Toista hakutyyppiä kutsutaan navigaatiohauksi (navigational searches) [GMM03].

Navigaatiohauksessa käyttäjän antaman hakuehdon tarkoitus on löytää määrätty ennalta tun-

nettu dokumentti. Käyttäjä saattaa esimerkiksi etsiä tieteellistä artikkelia käyttäen hakuehtona tämän otsaketta. Tällöin hakusana tai hakusanat eivät tarkoita mitään varsinaisia käsitteitä. Perinteinen sanahaku on navigaatiohakuihin luonnollisesti hyvä vaihtoehto, koska tällöin dokumentteja etsintään nimenomaan määrättyjen merkkijonojen perusteella [GMM03]. Tässä tutkielmassa keskitytään ensisijaisesti tutkimushakuihin.

Näkymäpohjaisessa haussa kysely muodostetaan valitsemalla käsitteitä ihmisen ajattelun kannalta luonnollisista käsitehierarkioista. Perinteisistä hakemistopalveluista (esimerkiksi DMOZ¹) poiketen haku voi koostua useista kohdetta eri näkökulmista kuvaavista käsitteistä. Käsitehierarkioiden myötä näkymäpohjaisessa haussa yhdistyvät hyvä löytyvyys ja tarkkuus iteratiiviseen hakuprosessiin. Perusmuodossaan hakuprosessi takaa tuloksellisen haun kaikissa tilanteissa ja tarjoaa mielekkäitä vaihtoehtoja haun muuttamiseen, laventamiseen ja tarkentamiseen. Myös hakutulokset esitetään tyypillisesti käsitehierarkioiden kontekstissa. Esimerkiksi sellaisten museoesineiden haku, jotka ovat valmistettu Suomessa, palauttaa kohteita ryhmiteltyinä lääneittäin. Haun tarkentaminen Etelä-Suomen lääniksi tarkentaa vastaavasti kohteiden ryhmittelyä kaupunkeihin. Toisaalta kohteet voi hakehdosta riippumatta ryhmitellä minkä tahansa käsitteen, kuten esinetyypin mukaisesti.

Vaikka monet käsitteet ovatkin luonnollisesti hierarkkisia, on myös paljon sellaisia käsitteitä, joilla ei ole luonnollista hierarkkista esitysmuotoa, tai sitten hierarkia on enemmänkin rajoittava kuin avartava. Esimerkki edellisestä on henkilön käsite. Varsinkin, jos kohteisiin liittyviä henkilöitä on paljon (vrt. puhelinluettelo), ei näiden hakeminen pitkältä sukunimen mukaan järjestetystä listasta ole mielekästä. Sen sijaan lomake, johon voi syöttää henkilön sukunimen, etunimen tai muita tietoja, on selkeästi parempi vaihtoehto. Hyvä esimerkki rajoittavasta hierarkiasta on aika (esimerkiksi valmistusaika). Ajalle on sinänsä luonnollinen hierarkkinen esitysmuoto (vuosisadat, vuosikymmenet, vuodet jne.), mutta tällainen jaottelu rajaa kuitenkin useita tyypillisiä aikahakuja pois. Esimerkiksi ennen tai jälkeen annetun vuoden, tai vuosisadan vaihteessa valmistettujen esineiden hakeminen ei tällaisella hierarkialla onnistuisi. Yleistäen literaalit, kuten vapaa teksti, numeeriset arvot, päivämäärät ja totuusarvot (*true/false*) istuvat huonosti hierarkian muotoon. Tällaisiin arvoihin kohdistuva haku on paras esittää lomakepohjaisesti.

Resource Description Framework (RDF) on *subjekti-predikaatti-objekti* -kolmikkoihin perustuva tietomalli [LS99]. Koska RDF-kieltä käytetään sekä itse datan että sen rakenteen eli skeeman tai ontologian kuvaamiseen, on näkymäpohjaiseen RDF-hakuun periaatteessa helppoa integroida alkuperäiseen skeemaan kohdistuvia osia. Perinteisten reaaliotietokantaan perustuvien näkymäpohjaisten hakukoneiden kannalta tämä tarkoittai-

¹<http://www.dmoz.org>

si lähinnä sitä, että hakukoneen oma tietokanta ja toteutus integroitaisiin kohdetietokantaan. Tyypillisesti (esimerkiksi Flamencossa [HEE⁺02]) näkymäpohjaiseen hakukoneeseen tuodaan hierarkioiden lisäksi vain kohteiden välttämättömimmät, ei-rakenteiset tiedot.

Datan tuottamisen ja ylläpitämisen kannalta RDF mahdollistaa sekä skeeman että datan käsittelyn yleisillä editoreilla. RDF-dokumentteja on mahdollista säilyttää tietokannassa tai sarjallistaa esimerkiksi tiedostoihin. RDF:n sarjallistamiseen on valittavana kolme yleistä sarjallistusmuotoa (RDF/XML, N3 ja NTriple), jotka soveltuvat eri tarkoituksiin.

Tässä tutkielmassa kuvattua näkymäpohjaista RDF-hakua on tutkittu Helsingin yliopiston tietojenkäsittelytieteen laitoksella, Semantic Computing -tutkimusryhmässä vuodesta 2002. Se on osa Eero Hyvösen kokonaisvisiota, MuseoSuomea [HJK⁺04], jossa yhdistyy

1. heterogeenisten tietolähteiden (esimerkiksi kaikkien Suomen museoiden aineistot) yhdistäminen ontologioiden avulla,
2. intuitiivinen näkymäpohjainen hakujärjestelmä, joka perustuu ontologisesti määriteltyihin yleis- ja yksilökäsitteisiin, sekä
3. semanttinen suosittelujärjestelmä, joka integroituu saumattomasti osaksi tätä kokonaisuutta.

Tutkielma jakaantuu rakenteellisesti kahteen osaan. Luvuissa 2–6 esitellään aiheeseen liittyvät yleiset käsitteet, paradigmat ja teknologia. Luvussa 7 esitellään näkymäpohjaista hakua ja määritellään näkymäpohjaisen RDF-haun yleiset vaatimukset. Luvut 8–9 perustuvat ensisijaisesti omaan tutkimukseen ja ratkaisumalleihin. Tutkielma vastaa kysymykseen, miten näkymäpohjaista hakuparadigmaa voidaan soveltaa ja edelleen laajentaa RDF-pohjaisesti?

MuseoSuomen ohella tutkielman esimerkeissä viitataan myös Helsingin yliopistomuseolle toteutettuun Promoottori-kioskisovellukseen, jonka aiheena ovat promootiokuvat [HSV04]. Promoottori on ensimmäinen tuotantokäyttöön otettu näkymäpohjainen RDF-hakukone.

Luvussa 2 määritellään mitä tarkoitetaan käsitteistöllä. Luku 3 määrittelee ontologian käsitteen ja sen suhteen sanastoihin ja kehyspohjaisiin kieliin. Luku 4 kuvaa yleisen nelitasoisen metatietoarkkitehtuurimallin. Luvussa 5 esitellään Semantic Webin kolme tärkeintä kieltä: *Resource Description Framework* (RDF), *RDF Vocabulary Description Language* (RDFS) ja *Web Ontology Language* (OWL). Luku 6 määrittelee, mitä tarkoitetaan

annotaatiolla, annotaatio-ontologialla ja ontologiapohjaisella annotaatiolla. Luku 7 esittelee näkymäpohjaisen haun yleisellä tasolla sekä miten se suhtautuu RDF-pohjaiseen tietämykseen. Luvussa esitellään myös RDF-kielen laajassa mittakaavassa mahdollistama ominaisuusrajoitelaajennus. Koska RDF-pohjaiset tietämuskannat eivät välttämättä sellaisenaan ole täysin yhteensopivia näkymäpohjaisen haun tietomallin kanssa, kohdistuu näkymäpohjainen RDF-haku datasta tehtyihin projektioihin. Projektioita käsitellään luvussa 8. Luvussa 9 esitellään osana MuseoSuomi-järjestelmää jo tuotantokäyttöönkin otettu näkymäpohjaisen RDF-hakumoottorin prototyyppi, Ontogator. Ontogator on yleinen näkymäpohjainen hakumoottori, jonka avulla on mahdollista toteuttaa hyvinkin erilaisiin käyttöliittymäratkaisuihin pohjautuvia näkymäpohjaisia hakukoneita. Luvussa 10 arvioidaan esitetyjä ratkaisumalleja sekä pohdiskellaan vaihtoehtoisia lähestymistapoja.

2 Käsitteistö

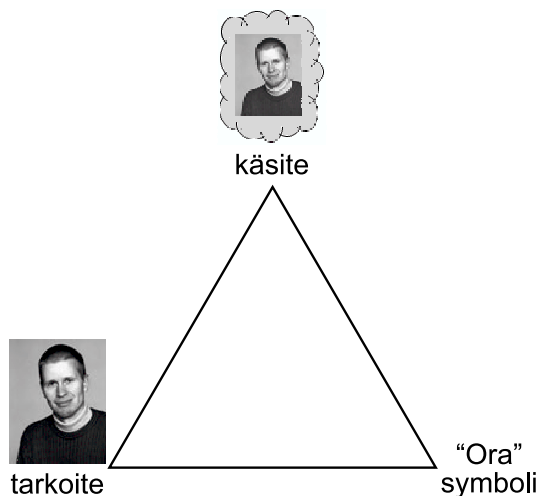
Käsitteistö koostuu useista käsitteistä, jotka on määritelty suhteessa toisiinsa. Tässä luvussa määritellään ensin mitä tarkoitetaan termillä *käsite* ja tämän jälkeen esitellään ajattelun ja ihmisten välisen viestinnän kannalta keskeisiä semanttisia suhteita.

Käsitteen määritelmä nojaa kahteen muuhun käsitteeseen, jotka ovat *tarkoite* ja *termi*. Tarkoitteet ovat joko konkreettisia (esineitä, ihmisiä tai eläimiä) tai abstrakteja (ominaisuuksia, tapahtumia, prosesseja). Käsitteet ovat ihmisen mielessään näistä muodostamia ajatuskokonaisuuksia. Käsitteet luokittelevat ja jäsentävät tarkoitteita. Ajattelun elementtinä käsitteet ovat olemassa vain ihmisen mielessä. Viestinnässä käsitettä voidaan käyttää vain, jos sillä on symboli, kuten termi. *Termi* on käsitteen kielellinen tunnus. Tarkoitteen, käsitteen ja symbolin suhdetta havainnollistaa *Ogdenin ja Richardsin kolmio* (kuvassa 1) [OR23].

Terminologiassa käsitteiden analysointi perustuu *käsitepiirteisiin* ja käsitteiden välisiin suhteisiin [TS888]. Käsitepiirre on sellainen tarkoitteeseen liittyvä ominaisuus, joka voidaan havaita tai mitata, tai joka yleisesti hyväksytään tarkoitteeseen kuuluvaksi. Käsitepiirteitä käytetään esimerkiksi käsitteiden luonnehtimisessa, rajaamisessa ja erottamisessa, samankaltaisuuden ja erilaisuuden vertailussa sekä määritelmien teossa.

Soveltavassa terminologiassa tiedon osien suhdetta kuvaa ehkä paremmin kuvan 2 tetraedrimalli. Tässä mallissa on huomioitu myös käsitteen kielellinen kuvaus eli *määritelmä*. Määritelmä

- yksilöi käsitteen niin, että se eroaa muista käsitteistä,



Kuva 1: Tarkoitteen, käsitteen ja termin välistä suhdetta havainnollistava Ogdenin ja Richardsin kolmio.

- määrittää käsitteen suhteet muihin käsitteisiin ja
- luo normit käsitteen käyttöä varten [TS888].

Formaali ontologia määrittelee käsitteistön tietojenkäsittelytieteellisessä mielessä. Ontologioita käsitellään luvussa 3.

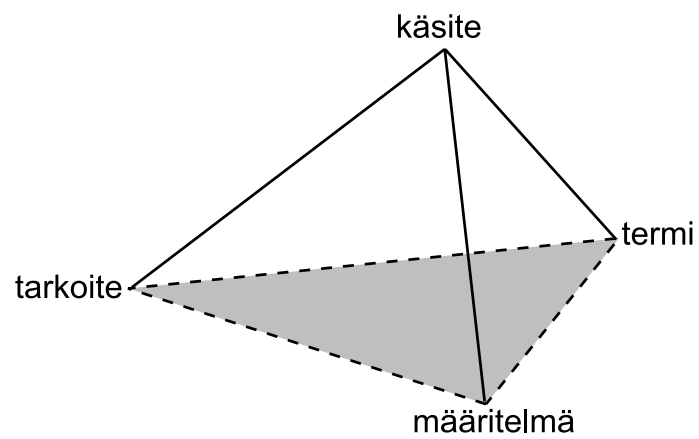
Jokaisella tarkoitteella on useita havaittavia ominaisuuksia, joista ajattelussa muodostetaan juuri tätä tarkoitetta vastaava *yksilökäsite* [TS888]. Yksilökäsite on siten yhtä tiettyä tarkoitetta vastaava käsite, joka ilmaistaan usein nimellä. Yksilökäsite vastaa oliomallinnuksen instanssia. *Yleiskäsite* syntyy, kun yksilökäsitteiden yhteiset ominaisuudet yhdistetään yhteiseksi käsitteeksi. Yleiskäsite vastaa oliomallinnuksen luokkaa.

2.1 Termien ja käsitteiden väliset semanttiset suhteet

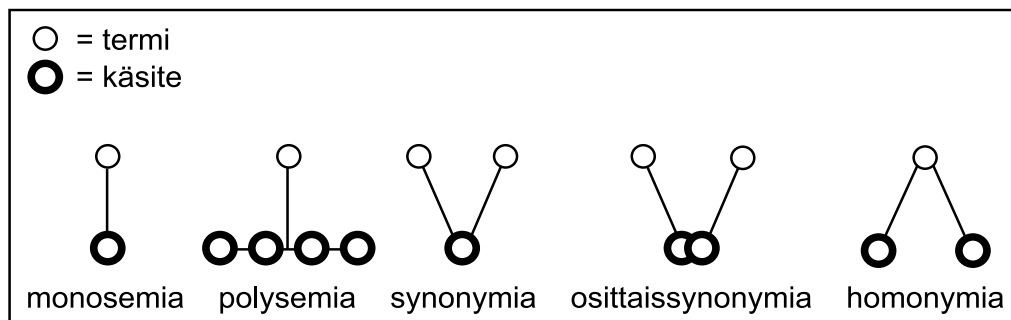
Käsitteiden ja termien välisiä suhteita ovat muun muassa monosemia, polysemia, synonyimia ja homonymia [TS888]. Kuva 3 havainnollistaa näitä alla kuvattuja suhteita.

Termin ja käsitteen suhde on *monoseeminen* eli yksikäsitteinen, jos termi vastaa vain yhtä käsitettä ja vastaavasti tätä käsitettä vastaa vain yksi termi. Monosemia on normatiivisen sanastotyön ihanne, mutta käytännössä monoseemit ovat hyvin harvinaisia.

Polyseeminen eli monikäsitteinen termi vastaa kahta tai useampaa käsitettä, jotka tavalla tai toisella liittyvät toisiinsa. Käsitteillä voi olla vähän yhteisiä piirteitä tai ne voivat perustua samaan kielikuvaan, mutta niiden yhtäläisyys on silti havaittavissa. Esimerkiksi



Kuva 2: Tarkoitteen, käsitteen, termin ja määritelmän välistä suhdetta havainnollistava tetraedri.



Kuva 3: Termien ja käsitteiden välisiä suhteita.

termi *pää* on polyseeminen ja vastaa käsitteenä ruumiinosaa tai työkalun osaa. Myös esimerkiksi termi *pyörä* on polyseeminen ja vastaa pyörää yleiskäsitteenä, polkupyörää sekä joissain yhteyksissä myös moottoripyörää.

Synonyymit ovat samaa käsitettä vastaavia termejä. Esimerkiksi termiä *pyörä* käytetään usein synonyyminä sanalle *polkupyörä*. *Osittaissynonyymit* ovat termejä, joita vastaavilla termeillä ovat samantapaiset tai lähes samat käsitepiirteet. Osittaissynonyymejä ovat esimerkiksi termit *talo* ja *rakennus*. Synonyymien huomioiminen parantaa haun kattavuutta.

Homonyymi on äänneasultaan (homofonia) tai kirjoitusasultaan (homografia) samanlainen termi, joka vastaa kahta tai useampaa toisistaan riippumatonta käsitettä. Esimerkiksi sana *kuusi* on homonyymi, joka asiayhteydestä riippuen tarkoittaa joko numeroa tai puulajia. Homonyymien erottaminen toisistaan parantaa haun tarkkuutta.

2.2 Käsitteiden väliset semanttiset suhteet

Käsitteiden (substantiivien) välisiä perustavanlaatuisia psykolingvistisiä suhteita ovat muun muassa hyponymia, meronymia ja antonymia. Psykolingvistiikka tutkii kieltä ajattelun ja viestinnän välineenä, puheen tuottamiseen ja ymmärtämiseen sekä kielen oppimiseen liittyviä kysymyksiä [Lei89]. Yleisessä terminologian teoriassa puhutaan myös funktiosuhteista [TS888].

Hyponymia tarkoittaa yläkäsite-alakäsite suhdetta. Tällä tarkoitetaan kahden käsitteen välistä epäsymmetristä transitiivista suhdetta, jossa toinen käsite on toisen alakäsite. Yläkäsitteellä (hypernym) ja alakäsitteellä (hyponym) on täsmälleen samat käsitepiirteet, minkä lisäksi alakäsitteellä on vähintään yksi lisäpiirre. Esimerkiksi käsitteellä *pyörä* on alakäsitteitä kuten *auton rengas* ja *vauhtipyörä*. Wierzbickan mukaan hyponymioita on viittä eri tyyppiä. Näistä tärkeimpiä ovat formaali ja funktionaalinen (telic) hyponymia. Formaali hyponymialla tarkoitetaan ”kana on lintu” -tyyppistä (IS-A-KIND-OF) suhdetta. Funktionaalinen hyponymia (IS-USED-AS-A-KIND-OF) puolestaan liittyy käsitteen käyttötapaan: ”kana on ruokaa” [Fel98].

Hyponymiahierarkiat ovat luonnollisia lähtökohtia näkymäpohjaisessa haussa käytettävälle käsitehierarkioille. Täsmällisyyden vuoksi on kyettävä kuitenkin erottamaan toisistaan erityyppiset hyponymiat. Olisi jokseenkin hämmentävää, jos ”kana lintuna” -haku palauttaisi kanakeiton reseptin tai vastaavasti ”kana ruokana” palauttaisi kanojen ruokintaan liittyviä dokumentteja.

Meronymia tarkoittaa transitiivista koostumussuhdetta. Meronymian huomioiminen haussa parantaa haun kattavuutta: koska Helsinki sijaitsee Suomessa ja Suomi Euroopassa, si-

jaitsee myös Helsinki Euroopassa. Meronymioita eli osa-kokonaisuus -suhteita on ainakin kuutta eri tyyppiä: komponentti-kokonaisuus (oksa/puu), jäsen-kokoelma (puu/metsä), osa-kokonaisuus (kakunpala/kakku), materiaali-objekti (alumiini/lentokone), vaiheprosessi (lapsuus/kasvaminen), paikka-alue (Helsinki/Suomi). Myös meronymiasuhteet ovat hyvä pohja hakunäkymille. Tällöin on kuitenkin muistettava, että vaikka meronymia on määritelmällisesti transitiivinen suhde, eivät eri tyyppiset meronymiat ole välttämättä ristiin transitiivisia. Tuntuu esimerkiksi hyvin omituiselta väittää, että oksa on osa metsää, vaikka metsä koostuukin puista ja puut oksista [Fel98]. MuseoSuomessa näkyvät valmistuspaikka ja käyttöpaikka ovat meronymisiä hierarkioita.

Antonymia tarkoittaa käsitteiden merkityksen vastakkaisuutta. Antonyymejä ovat esimerkiksi sanaparit kylmä-kuuma, hyvä-paha ja erilainen-samanlainen. Antonymian hyöty ei ole ilmeinen hakukoneen toteutuksen kannalta, mutta esimerkiksi semanttisen navigaattorin² tai suosittelujärjestelmän toteutuksessa siitä saattaisi olla hyötyä, sillä antonyymit esiintyvät luonnollisessa kielessä usein yhdessä. Antonyymit helpottavat myös käsitteiden merkityksen ymmärtämistä. Esimerkiksi kylmä on kuumen vastakohta.

Funktiosuhde on tavallinen oikeustoimen, tuotannon ja yleensä toiminnan käsitteiden välillä. Funktiosuhteisessa käsitejärjestelmässä saattaa kaikkien käsitteiden välillä olla eri suhde. Funktiosuhteita ovat esimerkiksi syyn ja seurauksen välinen suhde (syysuhde), ilmiöiden välinen seuraavuussuhde (tapahtumasuhde), tuottajan ja tuotteen välinen suhde (geneettinen suhde), toiminnan ja siinä käytetyn välineen välinen suhde (instrumentaalisuhde) sekä raaka-aineen ja siitä valmistetun tuotteen välinen suhde (tuotantosuhde).

Koska funktiosuhteet eivät välttämättä muodosta mitään luonnollista käsitehierarkiaa, on näiden käyttäminen käsitehierarkioiden muodostamisessa tapauskohtaista. Sen sijaan tavat, joilla haettavat kohteet liittyvät haettavaan käsitteisiin, ovat usein

funktiosuhteita. MuseoSuomessa näkyvät ovatkin nimetty funktiosuhteiden mukaisesti esinetyyppi-näkymää lukuunottamatta. Tällaisia näkymiä ovat esimerkiksi valmistuspaikka, valmistaja, käyttöpaikka, käyttötilanne ja käyttäjä. Nämä otsakkeet kertovat siitä suhteesta, mikä näkymään liittyvillä käsitteillä on haettavaan kohteeseen.

²Semanttisella navigaattorilla tarkoitetaan ohjelmaa, joka generoi linkkejä semanttisesti toisiinsa liittyvien dokumenttien välille.

3 Formaali ontologia

Semantic Web nojaa resurssien³ kuvauksessa ontologisesti määriteltyihin käsitteisiin. Tämän luvun tarkoitus on valaista käsitteen ontologia merkitystä ja täsmentää sitä, mitä sillä tarkoitetaan tässä tutkielmassa. Sanana ontologia on peräisin filosofisesta suuntauksesta, joka tutkii olemassaoloa kysymällä ”mitä todella on olemassa?” Tietojenkäsittelytieteen kontekstissa ontologialla tarkoitetaan formaalia eksplisiittistä käsitteistöä [Gru93].

Semantic Web -tutkimuksen ydin keskittyy *metaontologioihin* eli ontologioihin, joilla kuvataan ontologioita (ks. luku 4). Universaalien ontologian ohella Semantic Web -teknologioita hyödyntävissä tutkimuksissa puolestaan keskitytään usein alakohtaisiin (domain) ontologioihin. *Alakohtaisella ontologialla* tarkoitetaan jonkin yhteisön käsitystä rajattuun ongelma-alueeseen liittyvistä käsitteistä, näiden käsitteiden välisistä suhteista sekä niihin liittyvistä ominaisuuksista eli käsitepiirteistä. Se, mitä käsitteiden välisiä suhteita ontologian määrittelyssä on mahdollista käyttää riippuu käytettävästä ontologiakiielestä. Tärkein näistä suhteista on hyponymia.

Terminologisessa mielessä ontologia käsittää vain yleiskäsitteitä. Systemiä, joka käsittelee ontologian ohella myös yksilökäsitteitä kutsutaan *tietämyskannaksi* [Sow00, WSWS01].

3.1 Ontologia vai sanasto

Käsitteenä ontologia on varsin epämääräinen ja avoin. Se, mitä ontologialla täsmällisesti ottaen tarkoitetaan tai mitkä ovat ontologian vähimmäisvaatimukset jätetään ontologioita käsittelevissä dokumenteissa usein määrittelemättä. Implisiittisesti ontologian määritelmä ilmenee käytännössä siitä, mitä jätetään sanomatta tai mihin sitä verrataan. Formaaleista ontologioista puhuttaessa rajanveto sen suhteen mitä ontologia on ja mitä ei koskee usein käsitteiden *sanasto* ja *ontologia* välistä rajapyykkiä. Vaikka eri lähteiden käsitykset ontologian ja sanaston suhteesta poikkeavatkin toisistaan, ollaan yleisesti kuitenkin yhtämieltä siitä, että sanasto ei ole ontologia. Esimerkiksi artikkelissa *From Thesaurus to Ontology* [WSWS01] ontologialla tarkoitetaan jotain täsmällisempää ja semanttisesti rikkaampaa kuin mitä tesaurus⁴ on. Toisaalta kyseisessä artikkelissa ontologian esittämiseen käytetään RDFS-kieltä (ks. luku 5.2), joka on spesifikaation mukaan sanaston (vocabulary), ei ontologian määrityskieli. W3C ei sinänsä kiellä, etteikö RDFS-kieltä voisi käyttää

³Mikä tahansa tarkoite, jolla on URI-tunniste.

⁴Tesaurus on asiasanasto, johon sisältyy myös sanojen välisiä suhteita, kuten viittauksia laajempiin, suppeampiin ja rinnakkaisiin termeihin. Rinnakkaistermi eli assosiaatiosuhde tarkoittaa jonkinlaista löyhää käsitteiden välistä assosiaatiota, kuten ahdistuksen suhdetta eksistentialismiin, häpeään tai jännittämiseen.

ontologioidenkin kuvaamiseen, mutta varsinaisena ontologiakielenä W3C esittelee primitiiveiltään kuvauslogiikkaan (description logic) perustuvan OWL-ontologiakielen (ks. luku 5.3).

Sanaston määrittellessä termit, näiden väliset suhteet ja ominaisuudet suhteessa käsitteisiin, määrittelee ontologia itse käsitteet ja näiden keskinäiset suhteet. Ontologia ei välttämättä ota edes kantaa termeihin tai näiden välisiin suhteisiin, kuten synonymiaan tai homonymiaan. Myös automaattisesti luotua taksonomiaa voidaan pitää jonkin asteisena ontologiana. Psykolingvistikissa tai terminologisessa mielessä semanttisten suhteiden sijasta tällainen ontologia perustuu annetun dokumenttikorpuksen sanojen tilastollisiin suhteisiin. Tämänkaltaiset ontologiat eivät välttämättä ole ihmisen kannalta mielekkäitä itsessään tarkasteltuina, mutta esimerkiksi osana WWW-hakupalvelua tällaisen luokittelun hyöty on esimerkiksi hakutuloksen ryhmittelyn kannalta kiistaton. Useita näkökulmia (multi-faceted) käsittävien taksonomioiden automaattista muodostamista käsitellään esimerkiksi Buntinen, Pertun ja Tirrin artikkelissa *Building and Maintaining Web Taxonomies* [BPT02].

Jotta ontologia olisi sellaisenaan ihmisen kannalta mielekäs, tulee siihen liittyä termejä, ja käsitteiden välisten suhteiden tulee olla ihmisen ajattelun kannalta mielekkäitä (ks. luku 2). Esimerkiksi Princetonin yliopistossa kehitetty *WordNet*-ontologia⁵ määrittelee käsitteet termien avulla synonymijoukkoina⁶ [Fel98]. RDF(S)-kielessä käsitteet puolestaan identifioidaan URI-tunnisteilla ja mahdollinen käsitteeseen liittyvä termi (nimiö) määritellään *label*-ominaisuuden arvona (ks. luku 5). RDF(S)-kieleen pohajautuva OWL ei laajenna tätä tapaa käsitellä nimiöitä, joten myöskään OWL ei suoranaisesti ota kantaa luvussa 2.1 esiteltyihin terminologisiin suhteisiin.

Ontologia saattaa käsittää myös sellaisia käsitteitä, joille ei ole olemassa termiä. Jos esimerkiksi pyörä määriteltäisiin kulkuneuvon meronymiksi, ei reki voisi olla kulkuneuvo. Toisaalta pyörällisillä kulkuneuvoilla on kiistatta yhteisiä piirteitä (ainakin pyörät), joten käsitteen vaatimukset täyttyvät. Wordnet-ontologiassa tämä ongelma on ratkaistu välikäsitteellä ”pyörällinen_kulkuneuvo” [Fel98]. Monikielisissä ontologioissa⁷ jollekin käsitteelle voi olla termi yhdellä kielellä, mutta toisesta kielestä se saattaa täysin puuttua.

⁵Toisaalta voidaan kysyä, onko WordNet ontologia. WordNetin tekijät eivät kutsu sitä ontologiaksi vaan leksikaaliseksi (sanastolliseksi) tietokannaksi, eikä se pohjautu kehyspohjaiseen kieleen [Fel98]. Tästä huolimatta WordNetiä usein kutsutaan ontologiaksi (esim. Sowa 2000 tai Google-haku sanoilla WordNet ja ontology).

⁶Synonymijoukolla tarkoitetaan samaa tarkoittavien termien joukkoa.

⁷Monikielisellä ontologialla tarkoitetaan ontologiaa, jonka käsitteille on määritelty useita erikielisiä termejä.

Esimerkiksi Kiinan kielessä ei ole henkilöautoa vastaavaa termiä [Sow00, TS888].

Kivelä ja Hyvönen käsittelevät artikkelissaan *Ontological Theories for the Semantic Web* [KH02] ontologiakäsitteen eri tulkintoja sekä ontologioiden luokittelusysteemejä. Eräs ontologioiden luokittelussa käytetty piirre on formaalius. Ontologioiden määrittelyssä formaaliuden aste voi toteutuksesta riippuen vaihdella aina luonnollisesta kielestä tiukan formaaliin. Tiukan formaalilla kielellä määritellyt ontologiat ovat määritelmällisesti joko puoliformaaleja tai tiukan formaaleja. *Puoliformaalilla* ontologialla tarkoitetaan ontologiaa, joka on esitetty formaalisti määrittelyllä kielellä. *Tiukan formaalilta* ontologialta vaaditaan lisäksi perinpohjaisuutta, virheettömyyttä ja täydellisyyttä.

Koska Semantic Webissä käytetyn ontologiakielen on skaalautettava koko Webin mittakaavaan, joudutaan tiukan formalismin vaatimuksesta tinkimään. On täysin mahdollista, että kahdella toisistaan riippumattomalla taholla on täysin ristiriitaiset näkemykset samoista käsitteistä. Koska tällaisten mallien ratkeavuutta ei pystytä takaamaan, on OWL-kielestä kolme versiota. Näistä kahden (OWL Lite ja OWL DL) vaatimukset takaavat ratkeavuuden rajoissa pysymisen, mutta täydellä OWL-kielellä (OWL Full), samoin kuin RDF(S)-kielelläkin on periaatteessa mahdollista määritellä mitä vain [BvHH⁺04].

3.2 Kehykset

Vaikka esimerkiksi tietokantaskeemat ja olio-ohjelmat ovat muodollisesti täysin päteviä ontologioita, tarkoitetaan ontologialla tässä artikkelissa kehyspohjaisella kielellä esitettyjä deklariivisia määritelmiä. Marvin Minskyn vuonna 1975 kehittämä kehyksen käsite tarkoittaa jotain tyyppiä (type) tai tyyppin ilmentymää koskevien ominaisuuksien (slot) joukkoa [Sow00]. Minskyn artikkelin jälkeen erilaisia kehyspohjaisia kieliä on kehitetty jo yli viisikymmentä erilaista. Kehyspohjainen kieli määrittää kuvauksissa käytettävät primitiivit eli tyypit ja ominaisuudet. Tyyppi vastaa karkeasti terminologiateorian yleiskäsitettä ja tyyppin instanssi yksilökäsitettä.

Sowa käyttää kirjassaan *Knowledge Representation* [Sow00] kehyksien kuvaamiseen LISP-ohjelmointikielen syntaksia muistuttavaa notaatiota. Esimerkiksi henkilön käsitteen voisi määritellä kehyksenä seuraavasti:

```

1 (defineType      Human
2   (supertype    Thing)
3   (parent       (type Human)))
4 (defineType      Person
5   (supertype    Human)
6   (name         (type String))
7   (email        (type String)))

```

Tämä tarkoittaa ”Ihminen on tyyppi, jonka ominaisuuden vanhempi arvo on tyybiltään Ihminen. Henkilö on Ihminen-tyypin alityyppi, jolla on merkkijono-tyyppiset ominaisuudet nimi ja sähköpostiosoite”. Määritelmän ensimmäisellä ja neljännellä rivillä määritellään tyypit, *Human* ja *Person*. Toisella ja viidennellä rivillä määritellään näiden käsitteiden ylityypit eli hyperonyymit. Rivillä 3 määritellään tyyppille *Human* vanhemmuutta tarkoittava ominaisuus *parent*, joka voi saada arvokseen ihmistyyppiä olevan ilmentymän. Riveillä 5-7 määritellään tyyppin *Person* attribuutit (*name* ja *email*) sekä näiden tyypit (*String*). Tässä määritelmässä ominaisuudet *defineType* ja *superType* sekä tyypit *Thing* ja *String* ovat kielen primitiivejä. *Thing* tarkoittaa kaiken ylliluokkaa, jota logiikassa kuvataan merkillä *T*. Kuvassa 4 vasemmalla on tätä määritelmää vastaava UML-luokkakuvaus.

Ilmentymän tyyppistä *Person* voisi määritellä esimerkiksi näin:

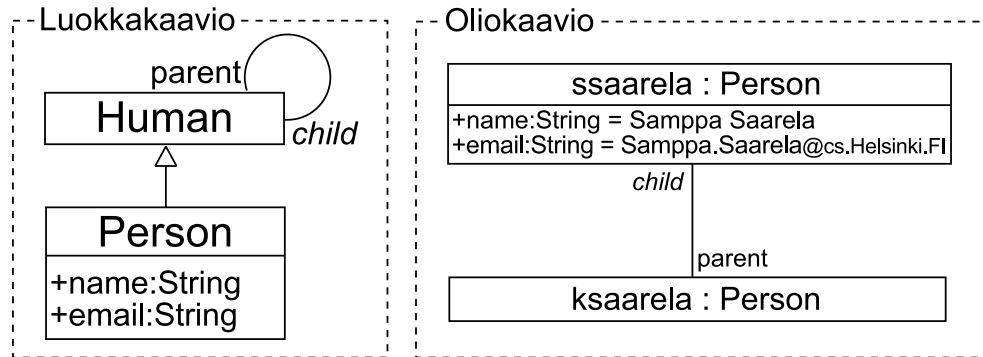
```

1 (defineInstance  ssaarela
2   (type          Person)
3   (name          'Samppa Saarela ')
4   (email         'Samppa.Saarela@cs.Helsinki.FI')
5   (parent        ksaarela))

```

Kuvauksen ensimmäisellä rivillä määritellään *ssaarela*-niminen instanssi, jonka tyyppi, *Person*, määritellään toisella rivillä. Riveillä 3-4 määritellään kyseisen henkilön ominaisuuksien *name* ja *email* arvot. Määrityksen viidennellä rivillä ominaisuuden *parent* arvo viittaa muualla määriteltyyn *Human*-tyyppiseen instanssiin *ksaarela*. Tästä esimerkistä ilmenee ylityypille määriteltyjen ominaisuuksien periytyminen alityypille: *Person*-tyyppisellä instanssilla on myös *Human*-tyypin ominaisuus *parent*. Kuvassa 4, oikealla on tämä kuvaus esitettyinä UML-oliokaaviona.

Kehyspohjaisen järjestelmän peruskäsitteistö vastaa oliopohjaisen ohjelmointikielen luok-



Kuva 4: Kuvassa vasemmalla UML-luokkakaavio tyyppejä Human ja Person vastaavista luokista ja oikealla kaksi luokan Person instanssia käsittävä oliokaavio.

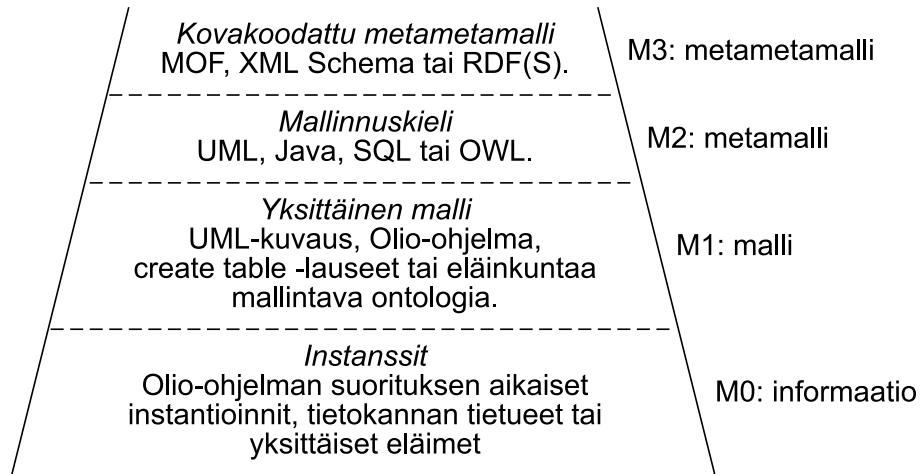
kamääriytyksien deklaratiiivisia osia. Eräs vaihtoehto ontologioiden mallintamiseen onkin UML-kielen luokka- ja oliokaaviot. Kuvassa 4 on yllä annettuja määritelmiä vastaavat tyyppiin Person ja Human sekä instanssien *ssaarela* ja *ksaarela* UML-määritelmät. UML-kielen käyttöä ontologioiden mallinnuksessa ovat tutkineet muun muassa Stephen Crane-field ja Martin Purvis [CP99].

Kehyspohjaiset järjestelmät tukevat luokkien moniperintää. Olio-ohjelmointiparadigmasta poiketen osa kehyspohjaisista järjestelmistä tukee moniperintää myös ilmentymien osalta eli moni-instantiointia: yksittäinen instanssi voi olla usean eri luokan ilmentymä [Sow00]. Myös RDF(S) ja OWL Full sallivat moni-instantioinnin. OWL-kielen Lite ja DL versioit kuitenkin rajoittavat tätä hieman. Esimerkiksi luokka ei voi olla samanaikaisesti ilmentymä tai ominaisuus.

4 Metatietoarkkitehtuurimalli

Tämä luku käsittelee meta-etuliitteen merkitystä ja käyttöä malleista (kielistä) puhuttaessa. Meta-käsitettä havainnollistaa klassinen nelitasoinen metatietoarkkitehtuurimalli. Kuvan 5 havainnollistaman metatietoarkkitehtuurimallin tasot ovat *metametamalli* (M3), *metamalli* (M2), *malli* (M1) ja *informaatio* (M0). Koska alemman tason mallin yksilö- tai yleiskäsitteet ovat ilmentymiä ylemmän tason tyypeistä, itse alemman tason malliakin kutsutaan toisinaan ylemmän tason mallin ilmentymäksi [Obj02].

Mallia, jota käytetään toisen mallin mallintamiseen, kutsutaan metamalliksi ja metamallin mallintamiseen käytettävää mallia vastaavasti metametamalliksi. Esimerkiksi UML-kielen luokkakaaviotekniikka on metamalli, jota käytetään muun muassa ohjelmien (varsinaisten mallien) kuvaamiseen. UML-mallin metametamalli puolestaan on *Meta Object*



Kuva 5: Nelitasoinen metatietoarkkitehtuurimalli.

Facility (MOF). Alin taso tässä nelitasoisessa metatietoarkkitehtuurimallissa on informaatiotaso [Sow00].

Erilaiset mallinnustarpeet vaativat erilaisia mallinnustyökaluja ja kieliä. Erilaisilla mallinnuskielillä on kuitenkin paljon yhteisiä piirteitä ja yleensä sama asia on mahdollista esittää useilla eri kielillä. Esimerkiksi Helsingin yliopiston tietojenkäsittelytieteen laitoksen kurssilla Tietokantasovellusten harjoitustyö tietokannan suunnittelu aloitetaan mallintamalla ongelma ensin UML-luokkakaaviotekniikalla. Tämän jälkeen luokkakaavio muunnetaan relaatiokaavioksi ja tämän pohjalta luodaan lopuksi SQL create table -lauseet. Metametamalli tarjoaa yhteisen pohjan eri kielillä esitettyjen mallien käsittelyyn.

Kovakoodattu *metametamalli* (taso M3) käsittää minimaalisen joukon korkean tason perustyyppjä, joita käytetään metatason mallien syntaksin ja semantiikan määrittämiseen. Perustyyppjä ovat esimerkiksi luokan käsite (*MetaClass*), ominaisuuden käsite (*MetaAttribute*) sekä ohjelmoinnista puhuttaessa yleinen operaation käsite (*MetaOperation*). Näiden lisäksi metametataso määrittää joukon yleiskäyttöisiä ensimmäisen kertaluvun tyyppjä kuten *Integer*, *Real*, *String* ja *Boolean* [Sow00].

Metamalli (taso M2) määrittää lisää mallinnuksessa käytettäviä tyyppjä kuten *Type*, *Instance*, *Class*, *Interface*, *Attribute*, *Association*, *Operation* ja *Enumeration*. Nämä tyypit ovat instansseja metametatason tyypeistä. Metamalleja ovat esimerkiksi UML, Java, SQL, OWL sekä luvun 3.2 esimerkeissä käytetty LISP-ohjelmointikieltä muistuttava kehyspohjainen kieli. Metamalli on tiettyä käyttötarkoitusta varten tarkoitettu mallinnuskieli [Sow00, Obj02].

Metatietoarkkitehtuurin toiseksi alimmalla (M1) tasolla on metamallin ilmentymänä kuvattu tapauskohtainen *malli*, kuten Tetris-pelin luokkakaavio tai lääketieteellinen ontolo-

gia. Malli on *metatietoa* kuvauksen kohteesta. Se jäsentää, luokittelee ja määrittelee eli käsitteistää varsinaista informaatiotasoa.

Varsinainen *informaatio* sijaitsee alimmalla M0-tasolla. Olio-ohjelmassa tämä informaatiotasoa aktualisoituu ajonaikaisesti. Tietokannoista puhuttaessa varsinaiset tietueet sijaitsevat tällä tasolla. Informaatiotasoa on määritelty metatietoarkkitehtuurissa nollassa tasolla sen vuoksi, että se ei tämän mallin mukaan ole enää metatietoa vaan varsinaista tietoa. Käytännössä yhden systeemin informaatio on kuitenkin toisen systeemin metatietoa ja esimerkiksi Semantic Webissä resurssien metatieto voidaan esittää tietämuskannassa, joka sisältää kaikkia(!) näitä tasoja. Semantic Web metatietokieliä käsitellään luvussa 5.

5 Semantic Web -metatietokielet

Semantic Web -metatietokielten juuret ovat tekoälyn ja logiikkakielten, erityisesti kehyspohjaisten kielten, tutkimuksessa [Hef01]. Tässä yhteydessä metatietokielten käsittely rajataan W3C-konsortiumin RDF-pohjaisiin kieliin. Luku 5.1 kattaa perustiedot RDF-metatietokielestä, luku 5.2 RDF Schema -sanaston kuvauskielestä ja luvussa 5.3 käsitellään tulevaisuuden Web-ontologiakieltä OWL, jonka ensimmäinen luonnos ilmestyi vuoden 2002 heinäkuussa ja suositus kuluvan vuoden (2004) helmikuussa. Kuvausten ei ole tarkoitus olla täysin kattavia ja kaikki piirteet ja ominaisuudet huomioonottavia, vaan lähinnä antaa yleiskuva kyseisistä kielistä ja niiden mahdollisuuksista.

5.1 Resource Description Framework (RDF)

Resource Description Framework (RDF) luo pohjan semanttisen (machine-understandable) metatiedon esittämiseksi Semantic Webissä. RDF-kielillä on kolme sarjallisuusmuotoa. Nämä ovat XML-pohjainen RDF/XML, ihmisystävällisempi ja tiiviimpi N3 ja koneen luettavaksi tarkoitettu N-TRIPLE, joka on olennaisesti vain lista kolmikoita [SW01].

RDF-kieltä voi käyttää sekä resurssien kuvaamisessa että uusien metamallien tai metamallien määrittelyssä. Esimerkiksi luvussa 5.2 kuvataan RDF-kielillä määritelty RDFS sanaston kuvauskieli, ja luvussa 5.3 tästä edelleen kehitetty ontologiakieli. On tärkeää huomata, että näin kuvatut kielet ovat syntaktisesti täysin RDF-yhteensopivia. Tämän RDF-kielen inkrementaalisen luonteen vuoksi ohjelma, joka törmää aikaisemmin tuntemattomalla RDF-pohjaisella kielellä kuvattuun määritelmään, pystyy kuitenkin käsittelemään sitä, vaikka se ei kaikkien kuvauksien semantiikkaa täysin ymmärtäisikään.

RDF-malli perustuu *subjekti-predikaatti-objekti* -kolmikkoihin, joiden voi ajatella muodostavan nimetyistä kaarista koostuvan suunnatun verkon. Näitä kolmikkoja kutsutaan lauseiksi (statement). Esimerkki 1 esittää erästä RDF-verkkoa. Verkon `rdf:Resource`-tyyppiset solmut (kuvassa 6 soikeat solmut) eli resurssit identifioidaan URI-tunnisteiden avulla, ja mikä tahansa mille voidaan osoittaa URI voi olla tällainen resurssi. Resurssi voi myös olla anonymi, jolloin siihen viittaaminen on mahdollista vain kyseisen dokumentin tai tietämuskannan sisältä käsin [LS99]. RDF-kielellä määritellyt suhteet ovat aina kahden solmun, subjektin ja objektin välisiä eli binäärisiä. Syntaktisesti RDF/XML-kielellä on mahdollista esittää useamman kuin kahden välisiä suhteita, mutta RDF-mallissa tämänkaltaiset suhteet palautetaan binäärisiksi.

5.1.1 Resurssien kuvaus

RDF-kielen XML-pohjaisessa esitysmuodossa lauseen subjekti ilmaistaan (perustapauksessa) *rdf:about* tai *rdf:ID*-attribuutilla. Anonymien solmujen tunniste määritellään *rdf:nodeID*-attribuutilla, jota käytetään myös näihin viitatessa. Esimerkissä 1 on esimerkki RDF/XML-kuvauksesta. Tässä resurssin kuvauksessa eli annotaatiossa määritellään Dublin Coren [WKLW98] metatietopredikaattien avulla artikkelin <http://www.cs.helsinki.fi/u/ssaarela/tiki/aine.ps> otsake (title) ja tekijä (creator). Annotaatiolla tarkoitetaan resurssien luettelointia eli metatietojen kirjaamista. Kuvassa 6 tämä annotaatio on visualisoitu verkkona. Kuva on tuotettu W3C-konsortion *RDF Validation Service* -ohjelmalla [RDF01]. Esimerkin 1 rivillä 9 määritellään kuvattavan resurssin tekijä, <http://www.cs.helsinki.fi/u/ssaarela>. Rivillä 11 määritellään, että kyseinen tekijä on `a: Person`-luokan instanssi ja riveillä 12–13, että hänen nimensä on *Samppa Saarela* ja sähköpostiosoite *Samppa.Saarela@cs.Helsinki.FI*. Sama kuvaus on annettu esimerkissä 2 N3-muodossa.

Tämän tutkielman leipätekstissä luokkien nimet kirjoitetaan isolla alkukirjaimella ja ominaisuudet pienellä. Luokat ja yleiset resurssit kirjoitetaan teletype-kirjasintyyllillä ja predikaatit kursivilla.

N3 on yksinkertainen, kompakti ja helposti luettava RDF-esitysmuoto [BLSC03]. Tässä tutkielmassa esimerkit annetaan ensisijaisesti N3-muodossa. N3:ssa lauseet annetaan muodossa

Subjekti Predikaatti Objekti .

Samaa subjektia koskevat lauseet päättyvät aina pisteeseen. Samaa subjektia koskevia lauseita voi niputtaa yhteen puolipisteellä:

 Esimerkki 1: RDF/XML-muotoinen annotaatio artikkelista aine.ps.

```

1 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2     xmlns:dc="http://purl.org/dc/elements/1.1/"
3     xmlns:a="http://www.ontologies.org/basic#">
4   <rdf:Description
5     rdf:about="http://www.cs.helsinki.fi/u/ssaarela/tiki/aine.ps">
6     <dc:title>
7       Ontologiapohjainen tietämyksen esittäminen Semantic Webissä
8     </dc:title>
9     <dc:creator rdf:resource="http://www.cs.helsinki.fi/u/ssaarela"/>
10  </rdf:Description>
11  <a:Person rdf:about="http://www.cs.helsinki.fi/u/ssaarela"
12    a:name="Samppa Saarela"
13    a:email="Samppa.Saarela@cs.Helsinki.FI">
14    <a:parent rdf:resource="http://www.edentities.com/ksaarela"/>
15  </a:Person>
16 </rdf:RDF>

```

```

S P1 01 ;
  P2 02 ;
  P3 03 .

```

Lauseet, joilla on sama subjekti ja predikaatti, voidaan myös yhdistää käyttämällä objektien välillä pilkkua:

```
S P1 01, 02, 03 .
```

URI-tunnisteet eli resurssit määritellään merkkien '<' ja '>' välissä tai nimiavaruus-etuliitteiden avulla. Literaalit määritellään lainausmerkkien välissä. Anonyymit resurssit määritellään merkkien '[' ja ']' välissä:

```
[ P1 01 ; P2 02, 03 ] .
```

Esimerkissä 2 riveillä 1–3 määritellään etuliitteet ja näitä vastaavat nimiavaruudet. Riveillä 4 ja 8 määritellään subjektit, joihin liittyvät ominaisuudet ovat vastaavasti riveillä 5–7 ja 9–12.

Esimerkki 2: N3-muotoinen annotaatio artikkelista aine.ps (sama kuin esimerkissä 1).

```

1 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2 @prefix dc: <http://purl.org/dc/elements/1.1/> .
3 @prefix a: <http://www.ontologies.org/basic#> .
4 <http://www.cs.helsinki.fi/u/ssaarela/tiki/aine.ps>
5   dc:title   ""Ontologiapohjainen tietämyksen esittäminen
6             Semantic Webissä"" ;
7   dc:creator <http://www.cs.helsinki.fi/u/ssaarela> .
8 <http://www.cs.helsinki.fi/u/ssaarela>
9   rdf:type  a:Person ;
10  a:name    "Samppa Saarela" ;
11  a:email   "Samppa.Saarela@cs.Helsinki.FI" ;
12  a:parent  <http://www.edentities.com/ksaarela> .

```

RDF-lauseen subjektin ja predikaatin tyyppi on aina `rdfs:Resource`. Objekti voi olla joko `rdfs:Resource` tai `rdfs:Literal`. Esimerkeissä 1, 2 ja kuvassa 6 objekteja ovat esimerkiksi literaali *Samppa Saarela* ja *dc:creator*-attribuutin `rdfs:Resource`-tyyppinen arvo `http://www.cs.helsinki.fi/u/ssaarela`.

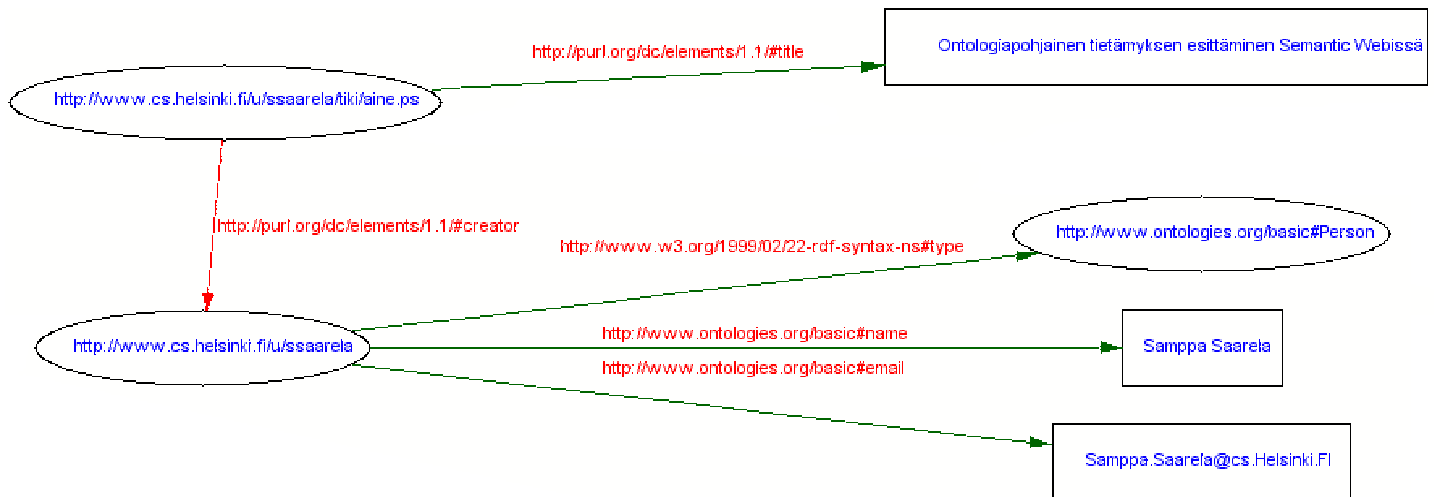
RDF ei ota kantaa literaalien tarkempaan tyypitykseen eli siihen onko kyseessä kokonaisluku, reaaliluku tai esimerkiksi totuusarvo. Koska RDF on XML-pohjainen kieli, on primitiivityyppien määrittäminen tarkoituksen mukaista XML Schema -kielellä. RDF-spesifikaation esimerkkejä mukaillen henkilön URI-tunnisteena käytetään hänen kotisivunsa `www`-osoitetta, mutta mikä tahansa muukin URI kävisi.

RDF käsittää sisäänrakennettuna myös listan (`rdfs:List`) ja erilaisten säilyttimien (Container) käsitteet. Säilyttimiä ovat esimerkiksi järjestetty `rdfs:Seq`, järjestämätön `rdfs:Bag` ja vaihtoehtoisia elementtejä sisältävä joukko `rdfs:Alt`. Säilyttimien elementit luetellaan *rdfs:_n* -predikaateilla, jossa $n \in \mathbb{N}$. Lista koostuu joukosta `rdfs:List`-tyyppisiä resursseja, joilla on lista-alkion arvoon osoittava ominaisuus *rdfs:first* ja seuraavaan lista-alkioon osoittava *rdfs:rest*. Listan päättää `rdfs:nil`. N3:ssa lista esitetään muodossa

(O1 O2 ... On).

5.1.2 Reifikaatio

RDF-kielen ilmaisuvoimaa lisää mahdollisuus tehdä lauseita, joiden subjektina on toinen lause. Tätä ominaisuutta kutsutaan *reifikaatioksi* (reification) ja sen avulla on mahdollista



Kuva 6: Esimerkkien 1 ja 2 RDF-kuvaus esitettyinä verkkona.

Esimerkki 3: RDF-esimerkki reifikaatiosta.

```

1 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2 @prefix dc: <http://purl.org/dc/elements/1.1/> .
3 @prefix a: <http://www.ontologies.org/basic#> .
4 [ rdf:type rdf:Statement ;
5   rdf:subject <http://www.cs.helsinki.fi/u/ssaarela/tiki/aine.ps> ;
6   rdf:predicate dc:creator ;
7   rdf:object <http://www.cs.helsinki.fi/u/ssaarela> ;
8   a:attributedTo <http://www.cs.helsinki.fi/u/jmanner> ] .

```

ilmaista metatietoa itse metatiedosta. Esimerkki 3 havainnollistaa reifikaatiota. Siinä on esimerkkien 1 ja 2 annotaatioesimerkkiin lisätty tieto annotaation esittäjästä: ”<http://www.cs.helsinki.fi/u/jmanner> väittää, että resurssin <http://www.cs.helsinki.fi/u/ssaarela/tiki/aine.ps> kirjoittaja on <http://www.cs.helsinki.fi/u/ssaarela>”.

Kuvan 3 esimerkki liittyy läheisesti visioon luottamuksen verkosta (Web of Trust) [Fen03]. Luottamuksen verkko koostuu henkilöistä, instituutioista, yrityksistä ja muista toimijoista, joilla on URI-tunniste. Ideana on, että kukin määrittää itse ne tahot, joihin luottaa. Jos henkilö *H* luottaa valtioon, joka puolestaan luottaa yliopistoonsa, niin tällöin myös *H* implisiittisesti luottaa kyseiseen yliopistoon. Liittämällä resurssiin tieto sen tekijästä ja allekirjoittamalla tämä väite digitaalisesti, on mahdollista automaattisesti päätellä voiko kyseisen resurssin sisältöön luottaa. Jo pelkästään ketjukirjeiden leviäminen ympäri maailman osoittaa, että tällainen luottamuksen verkko leviää hyvin laajalle, eikä sen täydellinen läpikäyminen ole välttämättä mahdollista riittävän tehokkaasti. Käytännössä verkot-

tuneiden metatietokuvausten käsittelyssä täytyy muutenkin turvautua tavalla tai toisella rajoitettuun laskentaan, esimerkiksi leveys- tai syvyysuunnassa rajoitettuun verkon käsittelyyn. RDF-verkon käsittelyssä tulee aina myös huomioida, että kuvaukset saattavat muodostaa syklejä.

Ongelmana RDF-kielen reifikaation esitystavassa on alkuperäisen reifikoidun lauseen tunnistaminen. Kuten kuvan 3 esimerkistä käy ilmi reifikoitavaan lauseeseen viitataan identifioimalla sen subjekti, predikaatti ja objekti. Tämä ei kuitenkaan riitä identifioimaan itse lausetta Webin mittakaavassa, sillä mikään ei estä etteikö samansisältöinen kolmikko voisi esiintyä eri yhteyksissä. Tällöin on epäselvää, mihin lauseeseen reifikaatio viittaa.

RDF ei itsessään sisällä kehyspohjaisille kielille ominaisia perustyyppisiä, kuten luokan käsitettä eikä mahdollisuutta määrittää resurssien välisiä hierarkkisia suhteita. Käytännössä RDF vaatii rinnalleen metametakielen, joka määrittää perustyyppit ja -ominaisuudet sekä tarjoaa välineet näiden välisten suhteiden määrittämistä varten. RDFS määrittelee minimaalisen joukon sanastojen määrittelemisessä tarvittavia perusprimitiivejä.

5.2 RDF Vocabulary Description Language (RDFS)

RDF-tason määrittäessä pohjan semanttisen metatiedon kuvaukselle, *RDF Vocabulary Description Language* eli RDF Schema (RDFS) luo pohjan RDF-kuvauksissa käytettävien semanttisten sanastojen kuvaamiselle. RDFS on tarkoitettu alimman tason ontologiakieleksi, joka käsittää vain välttämättömimmät primitiivit luokkien ja ominaisuuksien (attribuuttien) kuvaamiseen. Siinä missä kehyspohjaisissa kielissä puhutaan tyypeistä, puhutaan RDFS-kielessä UML-kielen tapaan luokista.

Tärkein RDFS-kielen määrittelemistä suhteista on hyponymia. RDFS mahdollistaa hyponymioiden määrittämisen luokkien (`rdfs:subClassOf`), instanssien (`rdf:type`) ja ominaisuuksien (`rdfs:subPropertyOf`) suhteen. Esimerkiksi luvussa 3.2 määritellyn tyyppin Human RDFS-version ominaisuudelle *parent* voisi määrittää adoptiovanhemmuutta tarkoittavan aliominaisuuden *adopter*. Koska adoptiovanhemmuus ei ole biologinen termi, olisi perusteltua rajata kyseinen ominaisuus koskemaan vain Person-tyyppisiä ihmisiä⁸. RDFS-kielessä on vain yhden tyyppistä hyponymiaa, joten se ei suoraan mahdollista psykologististen hyponymiasuhteiden IS-A-KIND-OF ja IS-USED-AS-A-KIND-OF erottelua.

RDFS-käsitteistö ei sisällä muita rajoitteita kuin ominaisuuksien kohteen (*rdfs:domain*) ja

⁸Adoptiovanhemmuus on sinänsä loogisesti ristiriitainen käsite biologisen vanhemmuuden kanssa, mutta esimerkiksi WordNetissä se on kuitenkin määritelty vastaavalla tavalla biologisen vanhemmuuden hyponymiksi (<http://www.cogsci.princeton.edu/~wn/>).

arvoalueen määrittymiset (*rdfs:range*). Kohde tarkoittaa luokkaa, jonka jäsenillä kuvattava ominaisuus voi olla. Arvoalue on kuvattavan ominaisuuden tyyppi eli se luokka, jonka instanssit sallitaan ominaisuuden arvoina.

Ominaisuuden *p* *rdfs:domain* määrittää siis niiden RDF-lauseiden sallitut subjektit, joissa *p* on predikaattina. Vastaavasti *rdfs:range* määrittää näiden lauseiden sallitut objektit. Jos ominaisuudelle määritellään useampia *rdfs:domain* tai *rdfs:range* määreitä, tulee subjektina/objektina olevan instanssin kuulua kaikkiin näihin luokkiin.

5.2.1 RDFS ontologiakielenä

RDFS ei ole ensisijaisesti tarkoitettu ontologiakieleksi vaan sanaston määrittämiskieleksi. RDFS-kielen ehkä suurin merkitys on kuitenkin siinä, että se toimii metametamallina yhdessä RDF-kielen ja XML Scheman kanssa OWL-kielelle. Tästä huolimatta sitä voi ja sitä myös käytetään ontologioiden kuvaamiseen. Esimerkissä 1 kaikki nimiavaruuksiin *rdf*, *dc* ja *a* viittaavat termit, kuten *a:Person*, ovat viittauksia RDFS-määritelmiin [BG04]. Esimerkki 4 on RDFS-kielellä määritelty ontologia, johon viitataan esimerkissä 1 *a*-nimiavaruudella.

Esimerkki 4 on kuvan 4 luokkakaaviota vastaava määrittymis RDFS-kielellä. Kuvan riveillä 5 ja 10 määritellään luokat *Human* ja *Person*. Riveillä 6 ja 11 määritellään näille suomenkieliset termit: *Human* → *Ihminen* ja *Person* → *Henkilö*. Näiden termien suomenkielisyys määritellään literaalilla seuraavalla *@fi*-määreellä. XML-notaatiossa kielen määrittämiseen käytetään *xml:lang*-attribuuttia. Rivillä 12 määritellään luokka *Person* luokan *Human* aliluokaksi. Riveillä 7–9 määritellään ominaisuus *parent* (vanhempi), jonka määritellään rivillä 8 olevan luokan *Human* jäsenillä ja rivillä 9 olevan arvoiltaan tyyppiä *Human*. Riveillä 13–18 määritellään *Person* luokan attribuutit *name* ja *email*.

Esimerkistä 4 käy ilmi RDFS-kielen tapa määrittellä ominaisuudet (alkaen riveiltä 7, 13 ja 16) erillään luokkamäärittelyistä (rivit 5–6 ja 10–12). Tämä mahdollistaa ontologian täyden laajennettavuuden missä päin Webiä hyvänsä. Kuvassa 4 on sama luokka esitetty UML luokkakuvauksena, jossa ominaisuudet määritellään sen luokan yhteydessä, jonka jäseniä ne ovat.

5.2.2 RDFS metametakielenä

RDFS-mallissa luokkamäärittelyt ovat luokan *rdfs:Class* instansseja —myös luokka *rdfs:Class* on tyypiltään *rdfs:Class* eli itsensä instanssi. Sekä luokka- että instanssi-

Esimerkki 4: RDFS-kielellä määritelty ihmisyyttä kuvaava ontologia.

```
1 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns> .
2 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
3 @prefix dc: <http://purl.org/dc/elements/1.1/> .
4 @prefix : <http://www.ontologies.org/basic#> .
5 :Human rdf:type rdfs:Class ;
6     rdfs:label "Ihminen"@fi .
7 :parent rdf:type rdf:Property ;
8     rdfs:domain :Human ;
9     rdfs:range :Human .
10 :Person rdf:type rdfs:Class ;
11     rdfs:label "Henkilö"@fi ;
12     rdfs:subClassOf :Human .
13 :name rdf:type rdf:Property ;
14     rdfs:domain :Person ;
15     rdfs:range rdfs:Literal .
16 :emain rdf:type :Property ;
17     rdfs:domain :Person ;
18     rdfs:range rdfs:Literal .
```

määrittelyt ovat RDF-mallin kannalta täysin samanlaisia (vrt. esimerkin 2 rivit 8–9 ja esimerkin 4 rivi 5). Luokkaa, jonka instanssit ovat itsekin luokkia, kutsutaan *metaluokaksi*. Vastaavasti ominaisuuksien määrittelyt ovat metaluokan `rdf:Property` instansseja. Alla kuvatut metaluokkien piirteet koskevat täten myös *metaominaisuuksia* [BG04, NFM00].

Metaluokat ja metaominaisuudet mahdollistavat uusien ilmaisuvoimaltaan vahvempien ontologiakielten määrittelemisen RDFS-kielellä. RDFS on täten *metaontologiakieli*, samoin kuin XML on rakenteisten dokumenttien *metakieli*. Näin voi esimerkiksi määrittellä luokalle `rdfs:Class` aliluokan `AbstractClass`, mallintamaan sellaisia luokkia, joilla ei ole instansseja. Tällaisissa määrittelyissä tulee kuitenkin olla hyvin varovainen, sillä näin määriteltyä ontologiaa tulkitsevan sovelluksen on tunnettava kyseisen metaluokan merkitys, pystyäkseen käsittelemään sen instansseja (abstrakteja luokkia) mielekkäällä tavalla. Toisaalta metaluokkia ja metaominaisuuksia on myös mahdollista määrittellä laajentamatta RDFS-kielen semantiikkaa. Näin on mahdollista määrittellä esimerkiksi synonymia: RDFS-kielessä resurssien symbolit (termi tai nimi, ks. luku 2) määrittellään ominaisuutena *rdfs:label*. Luomalla tälle aliominaisuus *synonym*, on mahdollista kuvata synonyymejä RDFS-yhteensopivalla tavalla. Synonymiaa syvällisemmin ymmärtämätön järjestelmä käsittelee näitä määrittelyitä *rdfs:label*-ominaisuuden tavoin, mikä kutakuinkin vastaa synonymian määrittelmää (ks. luku 2.1). Toisaalta synonymian ymmärtävät tahot pystyisivät huomioimaan synonyymit aivan erityisellä tavalla. Esimerkiksi kirjoitusohjelma voisi avustaa käyttäjää suosittamalla käsitteelle määriteltyä *rdfs:label*-määrettä huomattaessaan käyttäjän käyttävän tekstissään kyseisen käsitteen polyseemista tai homonyymista synonyymia. Synonyymien huomioiminen parantaa hakujen löytyvyyttä.

Rajanveto sen välillä, milloin metaentiteetti (metaluokka tai -ominaisuus) on sellainen metametatason määritelmä, joka vaatii ohjelmilta kovakoodatun käsittelijän, ja milloin yksinkertaisesti käytännöllinen mallinnusratkaisu, on usein vaikeaa. Metaentiteetti on metametatason määritelmä silloin, jos sen semantiikka poikkeaa olennaisesti käytettävän mallinnuskielen (esimerkiksi RDFS) semantiikasta. Esimerkiksi eri tyyppisiä transitiivisiä suhteita, kuten esivanhemmuutta tai meronymioita, ei ole mahdollista kuvata RDFS-kielen semantiikalla. Transitiivisiä ominaisuuksia varten tulisi määrittää metaominaisuus `TransitiveProperty`. Tätä ominaisuutta hyväksikäyttävien ontologioiden semantiikka ei kuitenkaan olisi yleisen RDFS-kieltä ymmärtävän sovelluksen tavoitettavissa. W3C-konsortiumin kehitteillä oleva OWL Web Ontology Language on määritelty RDFS-kielellä ja semanttiselta ilmaisuvoimaltaan huomattavasti RDFS-kieltä rikkaampi. Se käsittää muun muassa yllä visioidun `TransitiveProperty`-tyypin.

5.3 OWL Web Ontology Language

OWL Web Ontology Language [BvHH⁺04] on W3C-konsortiumin Semantic Web -ontologiakielisuositus. Semantic Web -ontologiakielen tavoitteet (goals), vaatimukset ja esimerkkejä käyttötapauksista on määritelty W3C-konsortiumin suosituksessa *Requirements for a Web Ontology Language* [OWL04]. Käyttötapauksina on esitelty muun muassa web-portaalit⁹ ja multimediatietokannat. Ontologiakielen tavoitteet liittyvät esimerkiksi generisyyteen, laajennettavuuteen ja käytettävyyteen. Evoluutioon ja yhteiskäyttöön liittyvät kysymykset ovat myös tärkeitä, sillä web-ontologiat ovat luonteeltaan muuttuvia ja hajautettuja. Kielen vaatimukset ovat konkreettisia ominaisuuksia (feature), jotka liittyvät yhteen tai useampaan määritellyistä tavoitteista. Esimerkiksi tavoite ontologioiden evoluution mahdollistamisesta johtaa vaatimukseen versionhallinnasta osana kieltä.

OWL pohjautuu *DARPA Agent Markup Language* [DAM] ja *Ontology Inference Layer* [HFB⁺00] kielten yhdistelmään, jota kutsutaan lyhenteellä DAML+OIL [DAM01]. OWL on määritelty RDFS-kielellä, joten RDFS on OWL-ontologioiden kannalta metametakieli. OWL laajentaa RDFS-käsitteistöä voimakkaasti predikaattilogiikan suuntaan mahdollistaen loogisen päättelyn ontologioiden perusteella. Tässä luvussa tarkastellaan OWL-ontologiakieltä luokkien, ominaisuuksien, instanssien ja ontologioiden määrittelemisen näkökulmista.

5.3.1 Luokkien määrittely

OWL-kielen määrittely aloitetaan määrittelemällä yleinen luokka `owl:Class`, joka on `rdfs:Class`-luokan aliluokka. Tämän jälkeen määritellään ontologian rajat: yleisin mahdollinen luokka `owl:Thing` (\top) ja sen komplementti `owl:Nothing` (\perp). Kaikki OWL-kielellä tehdyt luokkamäärittelyt ovat implisiittisesti luokan `owl:Thing` aliluokkia ja toisaalta luokka `owl:Nothing` on kaikkien luokkien aliluokka. Sillä ole yhtään ilmentymää.

OWL yleistää RDFS-kielen metaluokan käsitteen yleiseksi luokkakuvaukseksi (class description), joka voi aktualisoitua

- URI-tunnisteena (RDFS-kielen tapa määritellä luokkia),
- luokan jäsenet määrittävänä enumeraationa,
- ominaisuusrajoitteena,

⁹Yksi esimerkki ontologiapohjaisesta portaalista on OntoWeb, <http://www.ontoweb.org/>.

Esimerkki 5: OWL-kielellä viinin väriä kuvaavan piirteen määrittäminen luettelemalla kaikkien mahdolliset jäsenet [SWM04]. Luettelo annettu RDF-listana.

```

1 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2 @prefix owl: <http://www.w3.org/2002/07/owl#> .
3 @prefix : <http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#> .
4 :WineColor rdf:type owl:Class ;
5   owl:oneOf ( :White :Rose :Red ) .

```

- yhdisteenä kahdesta tai useammasta luokkakuvauksesta,
- leikkauksena kahdesta tai useammasta luokkakuvauksesta tai
- jonkin toisen luokkakuvauksen komplementtina.

Määriteltävän luokan instanssit on mahdollista luetella poissulkevasti *owl:oneOf*-ominaisuuden avulla. Esimerkissä 5 on viinin väriä kuvaava käsite määritelty luettelemalla sen mahdolliset jäsenet (valkoinen, rose ja punainen). Näin määritellyllä luokalla on täsmälleen luetellut jäsenet, ei enempää eikä vähempää [SWM04].

Luokka on myös mahdollista määritellä ehtona, jonka sen jäsenten tulee täyttää. Tällainen ehto koostuu ominaisuusrajoitteista (property restriction), joita ovat

- loogista kaikki-kvanttoria (\forall) vastaava *owl:allValuesFrom*,
- loogista on olemassa -kvanttoria (\exists) vastaava *owl:someValuesFrom*,
- ominaisuuden arvorajoite *owl:hasValue*,
- osallistumisrajoite eli kardinaliteetti (*owl:cardinality*, *owl:minCardinality* ja *owl:maxCardinality*).

Näitä rajoitteita käyttäen on esimerkiksi mahdollista määritellä, että luokka *Wine* on sellainen *PortableLiquid*, jonka valmistajista vähintään yksi on viinitila ja joka on valmistettu vähintään yhdestä rypälestä (Esimerkki 6).

Kolmas RDFS-kielen luokkien määrittäytapaa laajentava mahdollisuus on luokkien määrittely joukko-opillisilla operaattoreilla yhdiste (*unionOf*), leikkaus (*intersectionOf*) ja komplementti (*complementOf*). Nämä vastaavat loogisia konnektiiveja *disjunktio* (\vee), *konjunktio* (\wedge) ja *negaatio* (\neg). Lisäksi luokkien välisen ekvivalenssin voi määritellä *owl:equivalentClass*-ominaisuudella.

Esimerkki 6: OWL-pohjaisen viiniologian luokan Wine määritelmä ominaisuusrajoitteiden `someValuesFrom` (\exists) ja `minCardinality` avulla [SWM04].

```

1 :Wine rdf:type owl:Class ;
2   rdfs:subClassOf :PortableLiquid ;
3   rdfs:subClassOf [rdf:type owl:Restriction ;
4                   owl:onProperty :hasMaker ;
5                   owl:someValuesFrom :Winery ] ;
6   rdfs:subClassOf [rdf:type owl:Restriction ;
7                   owl:onProperty :madeFromGrape ;
8                   owl:minCardinality "1"] .

```

Esimerkki 7: Luokan WhiteWine (valkoviini) määritelmä: valkoviini on viiniä, jonka väri (`hasColor`) on valkoinen (White) [SWM04].

```

1 :WhiteWine rdf:type owl:Class ;
2   owl:intersectionOf ( :Wine
3                       [rdf:type owl:Restriction ;
4                       owl:onProperty :hasColor ;
5                       owl:hasValue :White ] ) .

```

Esimerkissä 7 on määritelty leikkauksen avulla luokka WhiteWine. WhiteWine luokan ilmentymät ovat sellaisia luokan Wine ilmentymiä, joiden ominaisuuden *hasColor* arvo on White.

Kuten RDFS-kielelläkin, on OWL-ontologiakielellä mahdollista määrittellä luokkia, jotka ovat useamman kuin yhden toisistaan poikkeavan luokan jäseniä. Luokkaan liittyvällä ominaisuudella *owl:disjointWith* voi määrittellä sellaiset luokat, joilla ei yhteisiä jäseniä voi olla. Näin olisi esimerkiksi mahdollista määrittellä, että viini ei ole mehua tai maitoa.

OWL nojaa yksinkertaisten tietotyyppien, kuten string, date, time, integer tai double, määrittelyssä XML Schema -kieleen. XML Schema -kieli käsittää kattavan joukon sisäänrakennettuja tietotyyppisiä ja mahdollistaa lisäksi uusien tietotyyppien määrittämisen. OWL ontologioista XML-tietotyyppisiin viitataan URI-tunnisteella. Esimerkiksi int-tietotyypin URI-tunniste on `http://www.w3.org/2001/XMLSchema#int`. XML Schema -kielellä määriteltyjä tietotyyppisiä käytetään `owl:DatatypeProperty`-tyyppisten ominaisuuksien arvoalueina (*rdfs:range*) [XML01, BvHH⁺04].

5.3.2 Ominaisuuksien määrittely

Luokkien kuvausmahdollisuuksien ohella OWL laajentaa merkittävästi RDFS-kieltä ominaisuuksien osalta. Ominaisuuksien määrittely pohjautuu olio- ja tietotyyppi-arvoisten ominaisuuksien erotteluun. Olioarvoinen (`owl:ObjectProperty`) ominaisuus saa arvokseen olioita ja data-arvoinen (`owl:DatatypeProperty`) saa arvokseen yksinkertaisten tietotyyppien ilmentymiä. RDFS-kielen kannalta yksinkertaiset tietotyypit ovat literaaleja.

OWL mahdollistaa ominaisuuksien yhtäläisyyden (*owl:equivalentProperty*) määrittämisen sekä käänteisten ominaisuuksien (*owl:inverseOf*) määrittämisen. Luvuissa 3.2 ja 5 käsiteltyä ihmisyyttä käsittelevää ontologiaesimerkkiä voisi edelleen kehittää, määrittämällä *parent*-ominaisuudelle käänteisominaisuus *child*. Tällöin tietämuskantaa käsittelevä ohjelma pystyisi kuvan 1 tietojen perusteella päättämään, että kssaarela-instanssilla on looginen ominaisuus *child*, jonka arvo on ssaarela. Käänteisominaisuus ja luokan määrittäminen komplementilla vastaavat yhden tyyppistä (tiukan loogista) antonymiaa.

Ominaisuuksien käyttöä on mahdollista rajoittaa RDFS-kielen *rdfs:range* ja *rdfs:domain* määreiden lisäksi osallistumisrajoitteilla, joukko-operaatioilla sekä kaikki- ja olemassalokvanttoreilla. Näitä ehtoja käytetään olio-ohjelmointitermein sanottuna luokkainvarianttien määrittämiseen ja niillä on kaksi selkeätä tarkoitusta: 1) toimia syötteenä päättelykoneelle, joka pystyy näiden perusteella muodostamaan uusia tosiasioita annetusta datasta ja 2) ohjata datan mielekkäässä käsittelyssä. Jälkimmäisessä roolissa nämä rajoitteet (ja ontologia ylipäänsä) toimivat esimerkiksi tietämuskantaeditorissa, ohjaamalla mielekkäiden määrityksien tekemisessä, estämällä esimerkiksi ettei ihmisen vanhemmaksi voi asettaa koiraa.

Rajoitusten lisäksi OWL sisältää neljä uudentyyppistä olioarvoista ominaisuutta:

- `owl:TransitiveProperty`,
- `owl:SymmetricProperty`,
- `owl:FunctionalProperty` ja
- `owl:InverseFunctionalProperty`.

`owl:TransitiveProperty` tyyppiset ominaisuudet (P) ovat nimensä mukaisesti transitiivisia, eli $P(x,y) \wedge P(y,z) \Rightarrow P(x,z)$. Tämä mahdollistaa esimerkiksi eri tyyppisten meronymioiden määrittelyn. `owl:SymmetricProperty` tarkoittaa symmetristä ominaisuutta (P) eli $P(x,y) \Rightarrow P(y,x)$. `owl:FunctionalProperty` tarkoittaa samaa kuin sellainen osal-

listumisrajoite, joka rajoittaa tietyn tyyppisen ominaisuuden määrän yhteen per instanssi, eli $P(x, y_1) \wedge P(x, y_2) \Rightarrow y_1 = y_2$.

5.3.3 Yksilöt

OWL-kieli mahdollistaa myös yksilöiden (informaatiotason olioiden) tarkemman määrittelyn. Koska RDFS-mallissa kaikki luokat (myös `rdfs:Class`) ja instanssit ovat tyyppiltään `rdfs:Resource`, ei RDFS-kielellä ole mahdollista määrittellä ominaisuuksia, joiden kohteena tai arvoalueena olisi nimenomaan yksilöitä (ei luokkia). `owl:Class` puolestaan ei itse ole tyyppiltään `owl:Thing` eikä sen aliluokka, mutta kaikki OWL-kielellä tehdyt luokkamääritykset ovat implisiittisesti `owl:Thing`-luokan aliluokkia. Näin esimerkiksi ominaisuudet, joiden *rdfs:domain* on `owl:Thing`, koskevat vain OWL-luokkien instansseja eli yksilöitä.

OWL ei myöskään oleta, että kahdella samaa yksilöä kuvaavalla objektilla olisi samat nimet (URI-tunnisteet). Instanssiin liittyvällä ominaisuudella *owl:sameAs* on mahdollista määrittellä kahden instanssin yhtäläisyys. Tälle vastakkaisella ominaisuudella *owl:differentFrom* voi puolestaan määrittellä kahden yksilön olevan erillisiä.

5.3.4 Ontologiat

Predikaattilogiikkaa muistuttavien luokkien ja ominaisuuksien määrittelytapojen lisäksi OWL-ontologiakielen kehityksessä on otettu huomioon ontologioiden evoluutioon ja yhdistämiseen liittyviä tekijöitä. Ontologia ja siihen liittyvät ominaisuudet identifioidaan luokan `owl:Ontology` avulla. `owl:Ontology`-luokan ominaisuus *owl:imports*-määrittää toisen ontologian, joka tulee sisällyttää kokonaisuudessaan osaksi määriteltävää ontologiaa. Ominaisuudella *owl:versionInfo* voidaan määrittellä ontologian versionhallintaan liittyviä tietoja.

Eräs suurimmista ontologioiden yhdistämiseen liittyvistä ongelmista on se, että kahden samaakin asiaa käsittelevän ontologian näkökulma ja rakenne voivat olla täysin erilaisia. Se, mikä yhdessä ontologiassa ilmaistaan instanssilla, voi toisessa ontologiassa ollakin luokka tai jopa ominaisuus. Vaikka aiemmissa OWL-kielen luonnoksissa *owl:sameAs*-predikaattia saattoikin käyttää minkä tahansa kahden resurssin samuuden kuvaamiseen, on varsinaisessa suosituksessa sen käyttö rajattu vain yksilöiden välille.

6 Annotaatio

Annotaatio on metatietoa annetusta kohteesta. Esimerkiksi valokuvan annotaatio voi käsitellä tietoja siitä milloin se on otettu, kuka sen on ottanut sekä mitä kuva esittää. Tässä luvussa käsitellään annotaatio-ontologian käsitettä sekä ontologiapohjaista annotaatiota.

6.1 Annotaatio-ontologia

Annotaatio-ontologialla tarkoitetaan annotaatioiden rakenteen ja semantiikan kuvausta. Se on ontologia siinä missä muutkin ontologiat ja sen erottaa näistä lähinnä käyttötarkoitus: annotaatio viittaa johonkin ulkopuoliseen objektiin ja sen ominaisuudet keskittyvät kuvaamaan tämän ominaisuuksia, eivät niinkään annotaation itsensä ominaisuuksia. Esimerkiksi annotaation tekijä-attribuutti ei tarkoita sitä, että kyseinen henkilö olisi annotaation tekijä vaan sen dokumentin, johon annotaatio viittaa. Toisaalta annotaatiolla voi olla omanakin ominaisuutena tekijä —tämä on metatietoa itse annotaatiosta. Siinä missä Semantic Webin toteutuminen laajassa mittakaavassa nojaa siihen, että samasta asiasta käytetään aina samaa URI-tunnistetta, voi yhdestä URI-tunnisteen osoittamasta kohteesta olla useita annotaatioita.

Yleistäen mitä tahansa resurssin kuvausta voidaan ajatella annotaationa: kaikki kuvaukset, kuten termit yleensäkin, viittaavat loppujen lopuksi johonkin ulkopuoliseen tarkoitteeseen. Haun kannalta annotaation voisikin määritellä haettavan kohteen kuvaukseksi. Esimerkiksi MuseoSuomessa haetaan museoesineitä. Museoesineitä on mahdollista hakea esimerkiksi käyttäjän ja valmistajan mukaan. Museoesineiden kuvaukset (luokan *Esinekortti* ilmentymät) ovat tällöin annotaatioita. Toisaalta näkökulmaa vaihtamalla haun kohteeksi voisikin määritellä *Henkilö*-luokan instanssit, joita haettaisiin muun muassa sen perusteella, mitä esineitä nämä ovat käyttäneet tai valmistaneet. Kysymys siitä, mikä on annotaatiota ja mikä ei riippuu siis viimekädessä näkökulmasta.

6.2 Ontologiapohjainen annotaatio

Usein, esimerkiksi Adoben metatietoratkaisussa¹⁰, metatietopredikaattien arvot annetaan tekstimuotoisina. Vaikka tällöinkin käytössä on annotaatio-ontologia, on kyseenalaista voiko ratkaisua kuitenkaan kutsua ontologiapohjaiseksi annotoinniksi. Annotaation attribuuteista tiedetään tällöin vain missä roolissa kyseinen merkkijono on suhteessa koh-

¹⁰<http://www.adobe.com/products/xmp/>

teeseen. Esimerkiksi tekijän osalta pelkkä nimi ei vielä yksilöi kyseistä henkilöä — eikä edes sitä onko kyseessä henkilö vai kenties yritys puhumattakaan siitä onko merkkijono ylipäänsä nimi vai mahdollisesti hieman paremmin tekijän yksilöivä sähköpostiosoite tai henkilötunnus. Tällä tavalla kuvattuja kohteita ei ole mahdollista hakea näkymäpohjaisella haulla, jossa lähtökohta on nimenomaan arvojen hierarkkiset määrittelyt. Tässä tutkielmassa ontologiapohjaisella annotaatiolla tarkoitetaan sellaista ratkaisua, jossa annotaatioissa käytettävä arvoavaruus on ontologisesti kuvattu.

Yksinkertaisimmillaan ontologiapohjainen annotaatio määrittelee yhden resurssiarvoisen ominaisuuden, jota käytetään viittaamaan ontologisesti määriteltyihin resursseihin (instansseihin). Näkymäpohjaisen haun kannalta tämä ratkaisu on täysin riittävä, selkeä ja yksinkertainen — esimerkiksi Promootori käyttää tätä annotaatiotapaa. Mahdollisen muun päättelyn, kuten suositusten kannalta tällaisten annotaatioiden käsittely on kuitenkin hankalaa: itse suhteen tyyppi ei ole ilmeinen. Siinä missä annotaatio-ontologiaan pohjautuva ei-ontologiapohjainen annotaatoratkaisu hukkaa arvon tyyppin, häviää arvon rooli suhteessa kohteeseen vastakkaisessa menettelytavassa.

Käytännössä yleiseksi tarkoitetuissa annotaatoratkaisuissa on kuitenkin hyvin vaikea elei mahdotonta määrittellä attribuuttien arvoalueita. Arvoalueiden määrittely kiinnittäisi samalla annotaatiossa käytettävän ontologian. Jopa sen luokan, johon attribuutit kuuluvat, määrittelemisen täysin yleispätevästi on hankalaa. Niinpä esimerkiksi Dublin Coren viidentoista metatietopredikaatin RDF skeema (<http://purl.org/dc/elements/1.1/>) ei ota kantaa näistä kumpaakaan.



7 Näkymäpohjainen haku

On olemassa useita erilaisia näkymäpohjaisiksi luokiteltavia hakurajapintoja. Hyvä historiikki ja yleisesittely erilaisista näkymäpohjaisista hakuratkaisuista löytyy osoitteesta <http://www.view-based-systems.com>. Kaupallisia toteutuksia ovat esimerkiksi <http://endeca.com> ja <http://www.siderean.com>. Informaatiohaun käyttöliittymätutkimuksen huippua edustaa Flamenco-järjestelmä [HEE⁺02, YSLH03]. Yhteistä näille kaikille on se, että haku muodostetaan joko kokonaan tai osittain valitsemalla kohteeseen eri näkökulmista liittyviä kategorioita. Museoesineisiin liittyviä näkökulmia ovat esimerkiksi esineen tyyppi, valmistuspaikka ja valmistusaika. Kuvassa 7 on MuseoSuomen aloitussivu, jolla näkyvät järjestelmän kaikki yhdeksän näkymää sekä ensimmäisen tason kategoriat.

Kategoriat muodostavat ortogonaalisia hierarkioita, joissa alikategoriat assosioituvat ylikategorioihinsa. Perinteisesti luokitteluun on käytetty tesauruksia, kuten *Art and Achi-*

MuseoSuomi - Microsoft Internet Explorer

File Edit View Favorites Tools Help

 **MuseoSuomi** 
- Suomen museot semanttisessa webissä -

Uusi haku | Ohjeet | Näytä kaikki kategoriat | Tietoa ohjelmasta | MuseoSuomi-palautte

Sanahaku:

Esimetyyppi (koko luokittelu)

[taideteokset](#) (115),
[aset ja ampumatarvikkeet](#) (59),
[asiat ja taloustarvikkeet](#) (410),
[henkilökohtaiset esineet](#) (159),
[julkisen tilan esineet](#) (23),
[kulkuneuvot ja kuljetusvälineet osineen](#) (97),
[koneet ja laitteet](#) (74),
[lämmitykseen käytettävät esineet](#) (4),
[muut esineet](#) (180),
[maatalous- ja karjanhoitovälineet](#) (4),
[puhtaanapitoon käytettävät esineet](#) (16),
[pukineet ja tekstiilit](#) (1801), [säilyttimet](#) (578),
[ulkokalusteet ja pihatarvikkeet](#) (4),
[urheilu- ja pehvälineet](#) (30),
[valaisuihin käytettävät esineet](#) (57),
[yhteisölliset esineet](#) (20), [käsityöt](#) (12),
[leikkikalut](#) (200), [pyyntivälineet](#) (35)
(näytä kaikki 23)

Materiaali (koko luokittelu)

[muut aineet](#) (88), [ruoka- ja nautintoaineet](#) (1),
[materiaalit](#) (3777), [rakennusaineet](#) (6)

Valmistaja (koko luokittelu)

[henkilöt](#) (867), [henkilöryhmät](#) (1),
[kaupungit](#) (14), [laitokset](#) (8),
[tuotemerkit](#) (122), [yhdistykset](#) (24),
[yhteisöt](#) (5), [yritykset](#) (1247)

Valmistuspaikka (koko luokittelu)

[Aasia](#) (35), [Afrikka](#) (116),
[Etela-Amerikka](#) (1), [Eurooppa](#) (2541),
[Pohjois-Amerikka](#) (10), [ulkomaat](#) (6)

Valmistusaika (koko luokittelu)

[aikakaudet](#) (3023), [vuosisadat](#) (3012)

Käyttäjä (koko luokittelu)

[henkilöt](#) (1774), [henkilöryhmät](#) (28),
[kaupungit](#) (4), [laitokset](#) (74),
[yhdistykset](#) (8), [yhteisöt](#) (20),
[yritykset](#) (76)

Käyttöpaikka (koko luokittelu)

[Afrikka](#) (3), [Eurooppa](#) (2232),
[Pohjois-Amerikka](#) (2)

Käyttötilanne (koko luokittelu)

[kulttuuritapahtumat](#) (16),
[kansalais-, harrastus- ja vapaa-ajantoiminta](#) (210),
[institutionaalinen toiminta](#) (15),
[juhlat ja seremoniat](#) (91),
[kohteelle tehtävät toimenpiteet](#) (559),
[kohteessa tapahtuvat muutokset](#) (120),
[muut tapahtumat](#) (12),
[maatalous ja karjanhoito](#) (82),
[ruoan- ja ruomanvalmistus](#) (47),
[sodat](#) (29),
[elollisten olentojen perustoiminnat](#) (146),
[einkaimot](#) (121),
[tutkimus- ja kehittämistoiminta](#) (20),
[urheilu](#) (57), [valmistustekniikat](#) (1587)

Kokoelma (koko luokittelu)

[Espoon kaupunginmuseon kokoelmat](#) (1190),
[Kansallismuseon kokoelmat](#) (1351),
[Lahden kaupunginmuseon kokoelmat](#) (1587)

Internet

Kuva 7: MuseoSuomen etusivu.

*ecture Thesaurus*¹¹ (AAT) tai luokittelujärjestelmiä kuten *Medical Subject Headings*¹² (MeSH) tai *ACM Computing Classification System*¹³ (CCS), jonka perusteella tämäkin dokumentti on luokiteltu. Myös psykolingvistisiin suhteisiin perustuvaa WordNet-sanastoa on käytetty esimerkiksi Flamenco-järjestelmässä [HEE⁺02]. Kohteet kuvataan suhteessa näihin kategorioihin.

Teknisesti näkymäpohjaisen haun erottaa muista hakutavoista se, ettei siinä haeta pelkästään kohteita vaan myös kategorioita, joiden perusteella hakua on mahdollista muuttaa, laventaa tai tarkentaa. Kategorioihin liitetään yleensä arvio siitä, kuinka monta kohdetta kyseiseen kategoriaan liittyy annetuilla hakuehdoilla. Peruseriaate on, että kategoriat, joihin ei liittyisi yhtään kohdetta, suodatetaan tavalla tai toisella pois. Kategorioiden suodatusta käsitellään luvussa 7.2.

7.1 Kohteiden haku

Näkymäpohjaisen haun perustapauksessa näkymät koostuvat kategoriahierarkioista ja kustakin näkymästä voi olla valittuna korkeintaan yksi kategoria. Kategoriaan liittyy kohteita joko suoraan tai alikategorioiden kautta. Hakutulokset käsittää ne kohteet, jotka liittyvät kaikkiin valittuihin kategorioihin. Hakuehtoa on myös mahdollista käsitellä rajoittavana. Tällöin hakutulokset käsittää kaikki kohteet, jos yhtään valintaa ei ole tehty.

Kuvassa 8 on Promoottorin hakusivu, jolta on valittu kaksi kategoriaa: Aktipäivä ja Yliopiston päärakennuksen Juhlasali. Promoottori visualisoi valitun kategorian kirkkaalla taustavärillä. Jotta käyttäjä ymmärtäisi valinnan koskevan myös alikategorioita Promoottori värjää myös näiden taustan samalla näkymäkohtaisella värillä kuin valitun kategorian. Hakutulokset näkyvät kuvan vasemmassa laidassa ja niihin on liitetty tieto siitä mitä hakuehtoon liittyviä kategorioita niissä tarkkaan ottaen on. Esimerkiksi listan ensimmäinen kuva liittyy aktipäivänä tapahtuvaan kunniatohtorien vihkimiseen ja juhlasaliin. Jos yhtään kategoriaa ei ole valittu, näyttää Promoottori hakutuloksena kaikki järjestelmän tuntemat kuvat.

Perinteisissä näkymäpohjaisissa hakuratkaisuissa näkymät määritellään erillään kategorioista ja valinta on aina tyypiltään kategoria. Näkymää itsessään ei ole mahdollista valita. MuseoSuomessa näkymät ovat kuitenkin itsekin (juuri)kategorioita, joten niiden valitseminen vastaa teknisesti minkä tahansa muun kategorian valitsemista. Tämä mahdollistaa

¹¹http://www.getty.edu/research/conducting_research/vocabularies/aat/

¹²<http://www.nlm.nih.gov/mesh/>

¹³<http://portal.acm.org/ccs.cfm>

HYM - Promoottori 1.2

:: Helsingin yliopiston museon promotionäyttö ::

Kuvasivu Haku sivu Ohjeet Promotion historiaa Tietoa ohjelmasta

Valitse kuva-aiheet esim. "Tapahtumat: Tanssitapahtumat" ja "Henkilöt, roolit ja instituutiot: Roolit"

Valinnat

Uusi haku - Taikenna hakua

Tapahtumat : Aktipäivä

Paikat : Juhlasali

Kuvat (109 kpl)

- > Kunniohtorien vihkiminen. > Juhlasali.
- > Tohtorien vihkiminen. > Juhlasali.
- > Maisterien vihkiminen. > Juhlasali.
- > Kunniohtorien vihkiminen. > Juhlasali.
- > Maisterien vihkiminen. > Juhlasali.

Sivu: 1 / 22 edellinen seuraava

Tapahtumat : Aktipäivä

Kuva-aiheen valinta | Tietoa valinnasta

- Sepeleensitojaispäivä (2)
- Aktipäivä (109)
 - Aktisalin menevään kulkueeseen järjestäytyminen (0)
 - Kutsuvieraiden saapuminen aktisaliin (0)
 - Kulkue aktisaliin (4)
 - Promoootioakti (106)
 - Kulkue juhlaohjelmienpalvelukseen (0)
 - Juhlaohjelmienpalvelus (0)
 - Kulkue yliopistolle (0)
 - Promoootiopäiväiset (0)

Henkilöt, roolit ja instituutiot :-

Kuva-aiheen valinta | Tietoa valinnasta

- Roolit (74)
 - Aikaisemmat yleiset sepeleensitojattaret (1)
 - Dekaani (2)
 - Gratista/gratisti (0)
 - Juhlamenojen ohjaaja (2)
 - Kansleri (3)
 - Kunniohtori (26)
 - Maisteri (9)
 - Pedelli (1)
 - Primusmaisteri (1)

Promoootit ja juhlaistunnot :-

Kuva-aiheen valinta | Tietoa valinnasta

- Juhlaistunnot (2)
- Promoootit vuosittain (107)
 - Promoootit 1600-luvulla (0)
 - Promoootit 1700-luvulla (0)
 - Promoootit 1800-luvulla (2)
 - Promoootit 1900-luvulla (93)
 - Filosofinen tiedekunta 30.5.1900 (0)
 - Filosofinen tiedekunta 30.5.1907 (1)
 - Filosofinen tiedekunta 31.5.1910 (0)
 - Filosofinen tiedekunta 29.5.1914 (0)

Paikat : Juhlasali

Kuva-aiheen valinta | Tietoa valinnasta

- Aukiot, tontit ja puistot (2)
- Rakennukset (190)
 - Helsingin tuomiokirkko (31)
 - Vanha ylioppilastalo (0)
 - Yliopiston päärakennus (140)
 - Aula ja Käytävät (13)
 - Juhlasali (109)
 - Opettajien lehtisali (0)
 - Opiskelijaruokala (0)
 - Parnasso (2)

Esineet, tunnukset ja kulkuneuvot :-

Kuva-aiheen valinta | Tietoa valinnasta

- Esineet (1)
- Kukat (4)
- Kulkuneuvot (0)
- Merkit, liput ja tunnukset (22)
 - Airutinauhat (7)
 - Akateemiset arvonmerkit (8)
 - Kunniamerkit (4)
 - Liput (0)
 - Sinetit ja vaakunat (0)
 - Virkatunnukset (5)

Ohjelmaesitykset, esittäjät, teokset ja tekijät :-

Kuva-aiheen valinta | Tietoa valinnasta

- Musiikkiesitykset, puheet ja runonlausunnat (17)
 - Musiikkiesitykset (6)
 - Aktipäivän musiikkiesitykset (6)
 - Juhlaohjelmienpalveluksen musiikkiesitykset (0)
 - Promoootioaktin musiikkiesitykset (6)
 - Promoootiokonsertti (0)
 - Retki- ja tanssiaispäivän musiikkiesitykset (0)
 - Sepeleensitojaispäivän musiikkiesitykset (0)
 - Puheet (11)
 - Runonlausunnat (0)

Kuva 8: Promoottorin haku sivu. Hakuehtona Aktipäivä ja Juhlasali

Kuva 9: MuseoSuomen hakusivu. Hakuehtoina hakusana ” espoo” ja kategoria *säilytysastiat*

uudentyyppiset käyttötapaukset, kuten sellaisten kohteiden hakemisen, joilla ylipäättään on valmistaja.

Kohteiden hakemisen kannalta näkymän käsite on mahdollista yleistää joukoksi jonkin ehdon tai ehtojen perusteella valittuja kohteita. Hakutuloksen kohteet saadaan leikkauksena¹⁴ eri näkyymiin liittyvistä kohteista. Esimerkiksi MuseoSuomi käsittelee jokaista erillistä hakusanaa omana näkymänään. Kuvassa 9 on hakusanaan ” espoo” pohjautuva näkymä. Hakusanalla haetaan sekä kategorioita, jotka näytetään näkymässä (esimerkiksi ”Valmistuspaikka > ... > **Espoo**”), että suoraan kohteita. Näin ollen MuseoSuomessa hakusanäkymään saattaa liittyä kohteita, vaikkei siihen liittyisikään yhtään kategoriaa.

¹⁴Joissain näkymäpohjaisen haun implementaatioissa on mahdollista valita leikkauksen ja yhdisteen väliltä.

7.2 Kategorioiden suodatus

Kategorioita suodatetaan sen perusteella, liittyykö niihin yhtään kohdetta annetuilla hakuehdoilla. Käytännössä suodatus on mahdollista visualisoida eri tavoin. Eri käyttötapauksia voi tukea paitsi poistamalla suodatetut kategoriat täysin näkyvistä myös harmaannuttamalla nämä. Harmaannuttamisella tarkoitetaan käyttökelvottoman käyttöliittymäkomponentin visualisoinnista siten, ettei käyttäjä yritä sitä turhaan painaa. Eräs museotyöntekijän kannalta olennainen käyttötapaus on sen selvittäminen, mitä kokoelmista vielä puuttuu. Tässä harmaannuttaminen on oiva apu. Toisaalta tavanomaista käyttöä harmaannuttaminen ei aina edistä: ruudun täytyessä harmaista kategorioista, valittavissa olevia kategoriota joutuu etsimään. Poistamalla suodatetut kategoriat saadaan näytöllä oleva tila käytettyä tehokkaammin hyväksi. Oli suodatus sitten visualisoitu miten vain, olennaista on estää tyhjiin hakutulokseen johtavat valinnat. Promoottori (kuva 8) harmaannuttaa suodatetut kategoriat. MuseoSuomi poistaa oletusarvoisesti suodatetut kategoriat (kuva 9), mutta mahdollistaa myös näiden harmaannuttamisen vaihtoehtoisena hakumoodina. Kuvassa 16 on muuten sama hakusivu kuin kuvassa 9 sillä erolla, että suodatetut kategoriat ovat harmaannutettu.

Siinä, missä hakumootorin kannalta suodatus tarkoittaa poistamista, on harmaannuttamisesta aina vastuussa käyttöliittymälogiikka. Kategoriat, joihin ei liity kohteita, harmaannutetaan. Käytännössä halutaan usein kuitenkin poistaa sellaiset kategoriat, joihin ei liity alunperinkään yhtään kohdetta. Tämä on hyödyllistä erityisesti silloin, jos käytössä on vain osia laajoista tai yleisistä ontologioista. Kuvista 10 ja 11 käy myös selkeästi ilmi poistamisen merkitys verrattuna harmaannuttamiseen käyttöliittymän kannalta: tyhjät kategoriat poistaen (kuva 10) koko luokittelu mahtuu kerralla näytölle, siinä missä harmaannuttaessa koko luokittelu (kuva 11) vaatii sivun vieritystä useiden näytöllisten verran.

Liittämällä kategorioihin tiedot sekä osumien määrästä että kohteiden kokonaismäärästä, voisi vastuun käytettävästä suodatustavasta jättää täysin käyttöliittymälle. Periaatteessa hakulogiikan ei siis tarvitsisi suodattaa kategorioita laisinkaan. Tällöin kuitenkin mahdollisesti jopa suurin osa vastauksesta olisi turhaa tietoa, joka on tarkoitettu piilotettavaksi käyttäjältä. Tehokkuuden kannalta vastauksen kokoa pienentävällä suodatustavalla on olennainen merkitys, joka entisestään korostuu kaistanleveyden rajoittamassa hajautetussa ympäristössä. Yleisen näkymäpohjaisen RDF-hakumootorin kannalta suodatusvaihtoehtoja on kolme:

1. Ei suodatusta -vaihtoehto jättää suodatettavien kategorioiden käsittelyn täysin käyttöliittymälogiikalle.

MuseoSuomi - Microsoft Internet Explorer

File Edit View Favorites Tools Help

HELSENKI INSTITUTE FOR INFORMATION TECHNOLOGY

MuseoSuomi

- Suomen museot semanttisessa webissä -

UNIVERSITY OF HELSINKI

Uusi haku | Takaisin hakusivulle | Ohjeet | Näytä kaikki kategoriat | Tietoa ohjelmasta | MuseoSuomi-palaute

Hakuehdot

Hakusana: *espoo* (ryhmittele kohteet) (poista)

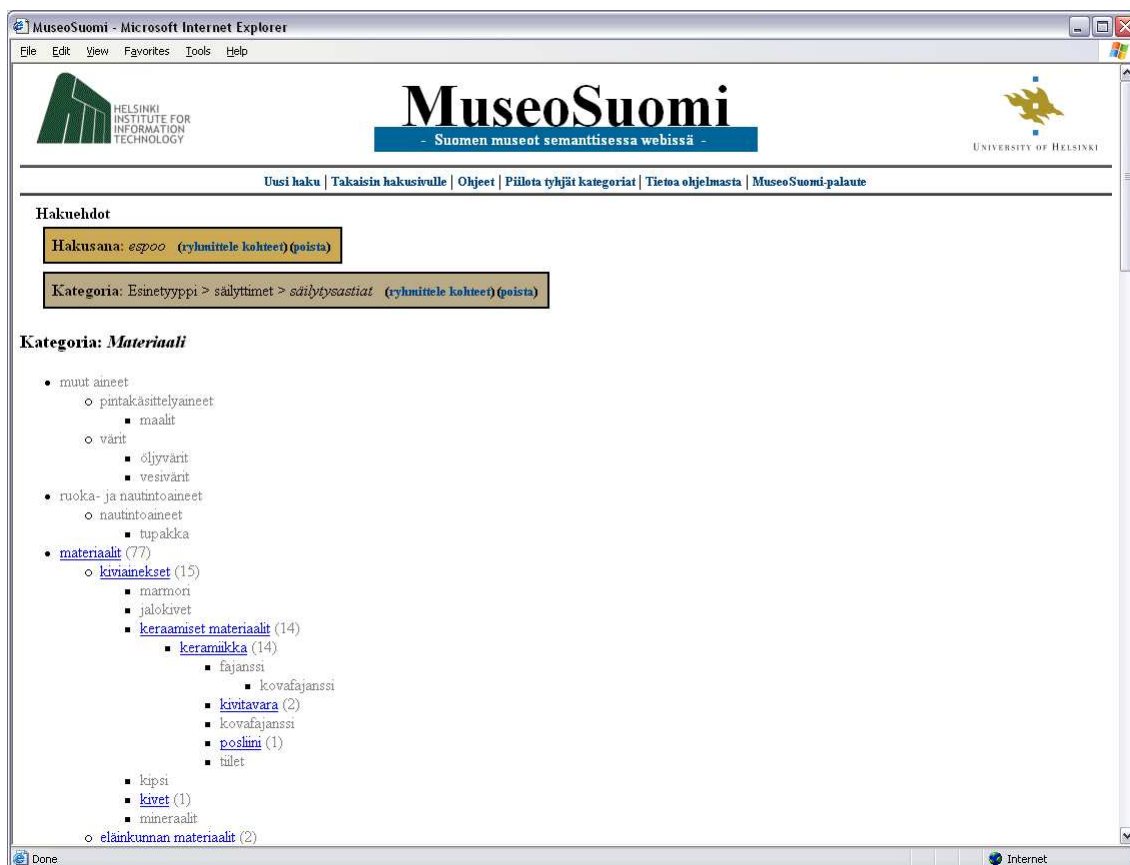
Kategoria: Esinetyyppi > säilyttimet > *säilytysastiat* (ryhmittele kohteet) (poista)

Kategoria: Materiaali

- [materiaalit](#) (77)
 - [kiviainekset](#) (15)
 - [keraamiset materiaalit](#) (14)
 - [keramiikka](#) (14)
 - [kivitavara](#) (2)
 - [posliini](#) (1)
 - [kivet](#) (1)
 - [eläinkunnan materiaalit](#) (2)
 - [nahka ja turkis](#) (2)
 - [nahka](#) (2)
 - [kasvikunnan materiaalit](#) (14)
 - [puu](#) (4)
 - [koivu](#) (2)
 - [kartonki](#) (7)
 - [pahvi](#) (2)
 - [paperi](#) (1)
 - [metallit ja metalliseokset](#) (21)
 - [metallit](#) (15)
 - [alumiini](#) (5)
 - [hopea](#) (8)
 - [peli](#) (6)
 - [muut materiaalit](#) (45)
 - [savi](#) (15)
 - [lasi](#) (21)
 - [muovi](#) (9)

Internet

Kuva 10: MuseoSuomen luokittelusivu. Hakuehtoina hakusana ”espoo” ja kategoria *säilytysastiat*.



Kuva 11: MuseoSuomen luokittelusivu. Hakuheutona hakusana ” espoo ” ja kategoria *säilyttimet*. Suodatetut kategoriat harmaannutettuina.

2. Ennen kyselyä tapahtuva suodatus poistaa sellaiset kategoriat, joihin ei alunperinkään liity yhtään kohdetta.
3. Kyselyn perusteella tapahtuva suodatus poistaa sellaiset kategoriat, joihin ei liity kohteita haun kontekstissa.

Kuvissa 16 ja 11 näkyvät harmaannetut kategoriat ovat kyselyn perusteella suodatettuja. Laventamalla kyselyehto tai aloittamalla uuden haun ne tulevat valittaviksi. Ei suodatusvaihtoehdossa käyttöliittymään saattaa tulla kategorioita, jotka ei koskaan ole mahdollista valita.

7.3 Kohteiden ryhmittely

Kategoriahierarkiat tarjoavat luonnollisen kontekstin haettujen kohteiden ryhmittelylle. 90-luvun puolivälissä kehitetty HiBrowse mahdollisti kohteiden ryhmittelyn samanaikai-

sesti usean eri näkymän suhteen [Pol97]. Käytetty analogia oli hakemistopuut, joissa dokumentit ovat lehtiä. Tavallisesta hakemistojärjestelmästä poiketen näitä hakemistopuita oli useita — yksi jokaista näkymää kohden. Hakemiston valinta yhdestä näkymästä poisti muista näkymistä sellaiset dokumentit, joihin kyseinen valinta ei liittynyt. Jos useita hakemistoja valittiin, saatiin leikkaus näihin kuuluvista dokumenteista. Promoottorin hakusivun hakemistojärjestelmän analogia on lainattu HiBrowsesta. Erona on se, että Promoottorissa hakutulokset esitetään listassa erillään hierarkioista.

Käyttöliittymätutkimuksen parista noussut Flamenco-järjestelmä esittää hakutulokset erillään hakuhierarkioista, mutta kuitenkin ryhmiteltyinä valitun hierarkian mukaisesti [HEE⁺02]. Oletusarvoisesti kohteet ryhmitellään viimeksi valitun kategorian mukaisesti, mutta järjestelmä mahdollistaa kohteiden ryhmittelyn myös minkä tahansa muun näkymän suhteen sen hetkisen valinnan perusteella. MuseoSuomessa hakuehto ja ryhmittelyehto ovat toisistaan riippumattomia ja ryhmittelyehtona on mahdollista käyttää myös hakusanaa. Esimerkiksi MuseoSuomessa sanahaku ” espoo ” palauttaa tulokset ryhmiteltyinä kategorioihin

- *Valmistuspaikka > ... > **Espoo**,*
- *Käyttöpaikka > ... > **Espoo**,*
- *Käyttäjä > ... > **Espoon kaupunginmuseo**,*
- *Kokoelma > **Espoon kaupunginmuseon kokoelmat**,*
- *Valmistaja > ... > *tekninen virasto* **Espoon kaupunki**,*
- *Käyttäjä > ... > *tekninen virasto* **Espoon kaupunki**,*
- *Valmistaja > ... > **Espoon partiotuki** ja*
- *Muita kohteita.*

Kategorian perusteella ryhmitellessään MuseoSuomi huomioi kyseisen kategorian lisäksi tämän mahdolliset alikategoriat (kuva 9). Hakusanan perusteella kohteet ryhmitellään vain löytyneiden kategorioiden mukaisesti huomioimatta mahdollisia aliluokkia. Sekä kategoria- että hakusanaperustaisessa ryhmittelyssä on huomioitava, että osa hakuehdon löytämisestä kohteista ei välttämättä kuulu mihinkään ryhmään. Tällainen tilanne voi syntyä, jos ryhmittelyperuste ei kuulu hakuehtoon tai hakuehto käsittää suoraan kohteisiin kohdistettuja hakuehtoja, kuten hakusanoja. Ryhmittelyyn kuulumattomat kohteet näytetään otsakkeen ”Muita kohteita” alla.

Kustakin ryhmästä näytetään tuloslistassa vain muutamia kohteita. MuseoSuomi sopeuttaa ryhmästä näytettävien kohteiden maksimilukumäärän ryhmien määrään: mitä enemmän ryhmiä sitä vähemmän kohteita kustakin ryhmästä. Sopeutus suoritetaan esikyselyllä, jonka vastauksena saadaan vain alikategorioiden (ryhmien) ja kohteiden lukumäärät. Jos ryhmiä on enemmän kuin kahdeksan¹⁵, sivutetaan tulokset ryhmien perusteella. Samoin mikäli kaikkia ryhmään liittyviä kohteita ei pystytä suoraan näyttämään, on niihin mahdollista päästä käsiksi hakuehtoa muuttamatta alisivutuksen kautta. Alisivutus on toteutettu ryhmien oikeassa alalaidassa olevilla edellinen/seuraava -linkeillä (kuva 9). Hakua tai ryhmittelyä on mahdollista edelleen tarkentaa myös ryhmien otsakkeista.

7.4 Näkymäpohjainen sanahaku

Näkymäpohjaisen haun ehkä suurin käytettävyysoongelma on se, että haun muodostaminen voi olla sekä hidasta että työlästä, jos hakukategoriat ovat syvällä kategoriahierarkioissa. Haun muodostaminen saattaa myös vaatia useiden hierarkioiden läpikäymistä, jotta halutut kategoriat löytyvät. Tämä ongelma korostuu erityisesti asiantuntijoiden käytötapauksissa. Tarkan hakuehdon kategoriat löytyvät luonnollisesti kategoriahierarkioiden lehdistä. Lyötään esimerkiksi MuseoSuomesta Espoon Kauklahdessa valmistetut maljakot, on käyttäjän joko etsittävä halutut kategoriat pitkistä koko hierarkian käsittävisistä listoista (ks. kuvat 10 ja 11) tai sitten navigoitava jopa kymmenen linkin kautta:

1. *Esinetyyppi: säilyttimet*
2. > *astiat*
3. > *säilytysastiat*
4. > *maljakot*
5. *Valmistuspaikka: Eurooppa*
6. > *Suomi*
7. > *Etelä-Suomen lääni*
8. > *Uusimaa-Nyland*
9. > *Espoo*

¹⁵Yhdellä hakusivulla näytettävien ryhmien maksimimäärä on konfiguroitavissa ja riippuu asiakkaan alustasta. Esimerkiksi mobiilipäätteelle ei mahdu yhtä monta ryhmää kuin tavanomaiselle näytölle.

10. > *Kauklahti*

Toinen ongelma liittyy haluttujen kategorioiden löytämiseen. Asiantuntijoiden laatimat hierarkiat eivät välttämättä avaudu tavalliselle käyttäjälle ja toisaalta myös asiantuntijoiden käsitykset käsitteiden merkityksistä poikkeavat toisistaan. ”Kumpi oli ensin, muna vai kana” -tyyppiset ongelmat ovat missä tahansa luokittelu- ja määrittelyongelmissa arkipäivää. Käytännössä todennäköisyys sille, että kaksi henkilöä valitsee saman termin kuvaamaan jotain tuttua käsitettä, on alle kaksikymmentä prosenttia [FLGD87].

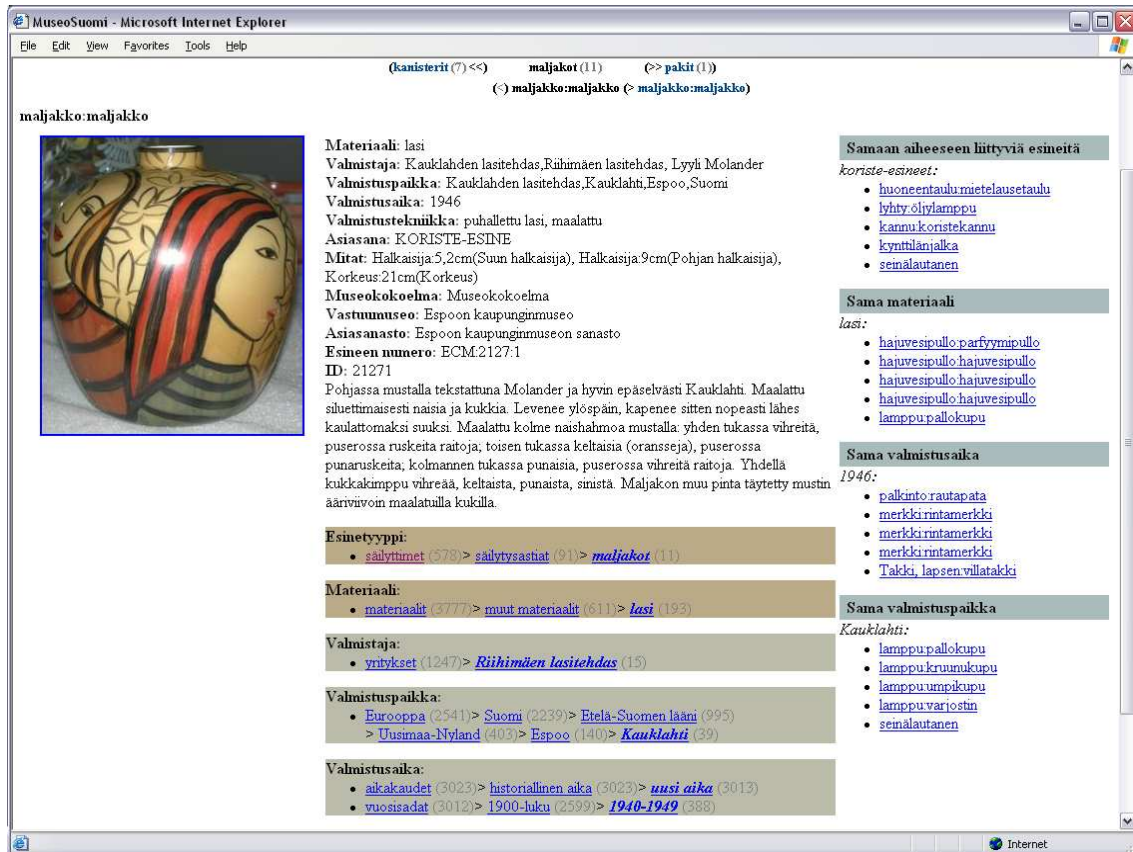
Luokittelun tuntemisen vaatimus on näkymäpohjaisessa haussa kuitenkin huomattavasti vähäisempi kuin perinteisissä hakusanoihin perustuvissa käyttöliittymissä: näkymäpohjaisessa haussa indeksointitapa on näkyvillä, siinä missä perinteisissä hakujärjestelmissä käyttäjä voi vain arvailla mitä termiä haettavassa dokumentissa tai sen indeksoinnissa on käytetty.

Kattavissa käytettävyytutkimuksissa on todettu, että aineiston ja sen indeksointitavat tuntevalle käyttäjälle suoraan sanahakuun (esimerkiksi Google) perustuvat käyttöliittymät ovat hyvä vaihtoehto tarkoilla kriteereillä haettaessa. [HEE⁺02, YSLH03]. Lavea tai hyvin yleiseen aiheeseen liittyvä haku koettiin kuitenkin ongelmalliseksi. Markkulan ja Sormusen kenttätutkimus painottaa aineiston selailun tarpeellisuutta [MS00].

MuseoSuomi yhdistää sanahaun saumattomasti näkymäpohjaiseen hakuun Flamenco vastavalla tavalla. Käyttäjän antamalla hakusanoilla etsitään paitsi itse kohteita myös kategorioita. Flamencosta poiketen kutakin hakusanaa käsitellään omana näkymänään vastaavaan tapaan kuin kategorianäkymiä. Tähän dynaamisesti luotuun näkymään kuuluvat kaikki sellaiset kategoriat, joiden nimessä kyseinen hakusana esiintyy. Näihin kategorioihin suorasti tai epäsuorasti liittyvien kohteiden lisäksi kyseiseen näkymään kuuluu sellaisia muita kohteita, joiden nimessä tai kuvaustekstissä annettu hakusana esiintyy. Kuvan 9 vasemmassa ylälaudassa on ’espo’-hakusananäkymä, johon liittyy annetuilla hakuehdoilla (hakusana ’espo’ ja kategoria *säilytysastiat*) kolme kategoriaa:

1. *Käyttöpaikka* > ... > *Espoo*,
2. *Valmistuspaikka* > ... > *Espoo* ja
3. *Kokoelma* > *Espoon kaupunginmuseon kokoelmat*.

Hakusananäkymät eroavat kategorianäkymistä siinä, että niihin kuuluvat kategoriat näytetään osittaisen polun kanssa. Polku näytetään, jotta käyttäjän olisi mahdollista erottaa homonyymit toisistaan. Kategorian valitseminen hakusananäkymästä poistaa kyseisen näkymän ja siirtää valitun kategorian siihen näkymään, johon se kuuluu. Jos tästä esimerkistä



Kuva 12: MuseoSuomen kohdesivu.

(kuva 9) valitaan kategoria ”*Käyttöpaikka > ... > Espoo*”, poistuu ’espo’-hakusananäkymä ja kyseinen kategoria siirtyy valituksi kategoriaksi *Käyttöpaikka*-näkömään.

7.5 Kohdesivu

Näkymiä voidaan hyödyntää myös kohteen tietojen esittämisessä kohdesivulla. Kun hakutuloksesta valitaan jokin kohde, siirrytään kohdesivulle. Kuvassa 12 on MuseoSuomen kohdesivu. Sekä MuseoSuomi että Flamenco näyttävät kohdesivulla kohteen perustietojen (kuvassa 12 keskellä) lisäksi kategoriasuhteet (kuvassa 12 keskellä alhaalla). Kohteeseen liittyvistä kategorioista näytetään niiden täysimittaiset polut. Jokainen polkujen kategorioista on linkki, joka muodostaa uuden haun. Tällä ominaisuudella pyritään tukemaan käyttäjän työkulkua (workflow) ja kiinnostuksen kohteen vaihtumista mahdollisimman tehokkaasti ja läpinäkyvästi [HEE⁺02].

Kohdesivulla näytetään myös hakutulosta läpikäyvät *edellinen* ja *seuraava* -linkit. Nämä linkit vievät sen ryhmän sisällä seuraavaan kohteeseen, mistä kohdesivulle alunperin siir-

ryttiin. Tämän lisäksi MuseoSuomi mahdollistaa myös vastaavalla tavalla navigoimisen hakutuloksen ryhmien välillä.

MuseoSuomessa kohdesivulla esitetään myös tietämuskantaan perustuvia suosituksia. Tällaisia suosituksia ovat esimerkiksi samaan aihepiiriin liittyvät kohteet. Esimerkiksi Kalevala-aiheisen veistoksen suosituksiin kuuluu kohteita, jotka liittyvät karelianismiin, kuten perinnepukuja.

7.6 Ominaisuusrajoitteet

Tässä aliluvussa käsitellään ominaisuusrajoitteisiin perustuvia näkymiä. Ominaisuusrajoitteiden ideana on, että esimerkiksi Henkilö-näkymän voisi esittää teennäisen sukupuoleen tai sukunimen alkukirjaimen perustuvan jaottelun sijasta lomakkeena, johon on mahdollista syöttää henkilön ominaisuuksiin kohdistuvia ehtoja. Tämä on tyypillinen tapa, jolla tietokantapohjaisten järjestelmien hakuliittymät on toteutettu: jokaista taulun saraketta kohden on lomakkeessa kenttä, johon hakuehto syötetään. Tulokseksi saadaan sellaiset taulun rivit, joilla kaikki annetut ehdot toteutuvat. Kentän jättäminen tyhjäksi tarkoittaa, ettei kyseistä saraketta huomioida.

Semantic Web mahdollistaa lomakepohjaisen haun yhdistämisen näkymäpohjaiseen haakuun tavalla, joka olisi perinteisellä tietokantapohjaisella lähestymistavalla mahdollista vain, jos näkymäpohjainen hakujärjestelmä integroitaisiin saumattomasti kohdetietokantaan. Koska sekä alkuperäinen skeema että itse data esitetään RDF-kielellä, on esimerkiksi mahdollista luoda automaattisesti annettua luokkaa vastaava hakulomake. Sinänsä tämä on mahdollista, mutta ei läheskään yhtä suoraviivaista, relaatiotietokantapohjaisella lähestymistavalla.

Ontogatorin (MuseoSuomen hakumoottori) käyttämät kategoriat projisoidaan ontologioista ja tietämuskannoista (ks. luku 8). Koska sekä alkuperäinen data että kategoriaprojektiot esitetään RDF-kielellä, integroituvat ne yhteisten URI-tunnisteiden ansiosta saumattomasti yhteen. Tällöin kategorioiden ja alkuperäisen datan välisissä viittauksissa ei tarvitse huomioida vaihtelevia taulujen ja näiden rivien tunnistusmenetelmiä (id-kentät). Käytännössä perinteisen tietokantapohjaisen lähestymistavan kannalta vielä suurempi ongelma onkin se, että yhtä kategoriata saattaa vastata relaatiotietokannan taulu ja tämän alikategorioita taulun rivit, tai sitten alikategoriat esitetään erillisinä tauluina ja instanssit riveinä. Vastaavan toiminnallisuuden aikaansaamiseksi järjestelmään tulisi täten implementoida tapa viitata itse taulumääritelmiin. RDF-maailmassa puolestaan kaikella, niin luokilla kuin instansseillakin, on URI-tunniste, johon on helppoa viitata.

Kategoriaprojektiot viittaavat alkuperäisen tietämuskannan kohderesursseihin *ogt: projectionOf*-predikaatilla. Sama resurssi voi moniperinnän ja eri näkökulmien vuoksi (esimerkiksi hyponymia ja meronymia) esiintyä useissa eri kategoriahierarkioiden haaroissa. Ontogator mahdollistaa kategorioiden hakemisen tämän predikaatin peruteella. Tätä ominaisuutta hyödynnetään esimerkiksi MuseoSuomen mobiiliversiossa: paikannuspalvelulta saadaan käyttäjän sijainnin perusteella paikan URI-tunniste, jota käytetään yhden näkymän luomiseen vastaavaan tapaan kuin hakusanoja (ks. luku 7.4). Tällä hetkellä paikkaan liittyviä näkökulmia on kaksi: valmistuspaikka ja käyttöpaikka.

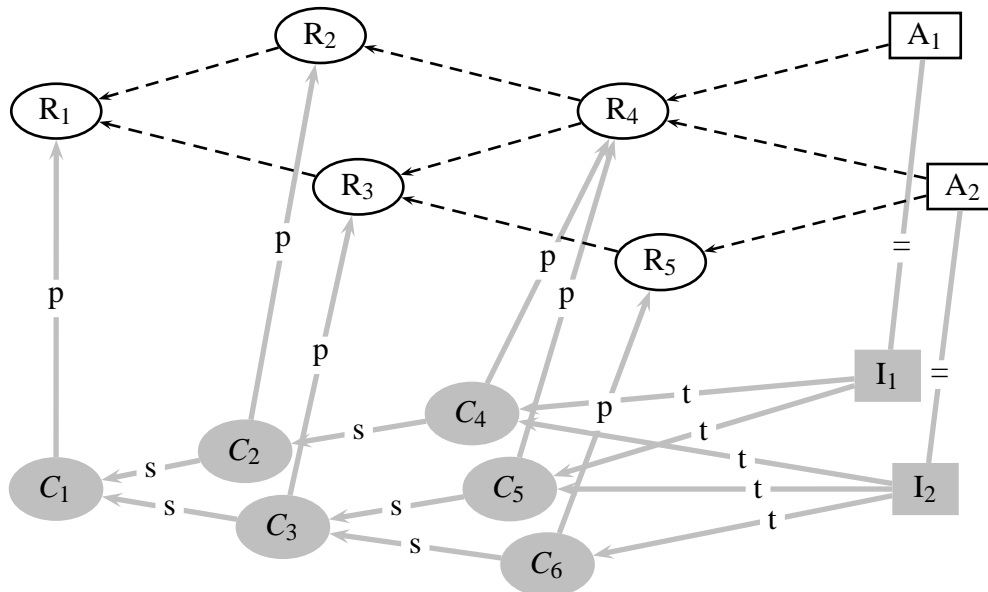
Kategorioiden hakeminen projektion perusteella mahdollistaa myös suositusten tiukemman integraation hakukoneeseen. Koska suositukset perustuvat tietämuskannan resursseihin, voitaisiin suositusten perusteita käyttää edelleen hakuehtoina näkymäpohjaiselle hakukoneelle. Esimerkiksi kuvan 12 ”Samaan aiheeseen liittyviä esineitä”-suosituksen (oikealla ylhäällä) kohta ”koriste-esineet” voisi olla linkki, joka muodostaisi uuden haun kyseisen resurssin perusteella. Tällöin ei kuitenkaan olisi mahdollista taata epätyhjää hakutulosta ilman suositusten esikäsittelyä.

Myös sanahaku olisi mahdollista käsitellä ominaisuusrajoitteena. Nykyisessä toteutuksessa ne predikaatit, joihin sanahaku kohdistuu, määritellään konfiguraatiossa. Tulevaisuudessa tämänkin voisi määritellä osana kyselyä, yhdenlaisena ominaisuusrajoitteena.

8 Näkymien projisointi

Tässä luvussa käsitellään näkymien projisointia. Näkymäprojektiot ovat näkymäpohjaisen haun tietomallin mukaisia projektioita haun kohteena olevasta tietämuskannasta. Niitä käytetään yksinkertaistamaan hakuprosessia sekä yleistämään hakumoottorin implementaatio. Näkymien projisointi on mahdollista suorittaa joko täysin dynaamisesti tai muuten ajonaikaisesti hakukoneeseen integroidulla päättelykoneella tai etukäteislaskennalla. MuseoSuomessa näkymäprojektiot tuotetaan etukäteen käyttäen Prolog-logiikkakohjelmointikieltä. Projektion tulos, eli kategoriahierarkioiden ja kohteiden määrittelyt, tallennetaan RDF/XML-muodossa tiedostoon, joka sitten ladataan MuseoSuomen hakumoottoriin, Ontogatoriin.

Kuva 13 havainnollistaa projektion ja alkuperäisen datan suhdetta. Kuvassa valkoiset resurssit (R_n), annotaatiot (A_n) ja kaaret kuuluvat alkuperäiseen tietämuskantaan, josta harmaat kategoriat (C_n), kohteet (I_n) sekä näiden väliset suhteet ovat projisoitu. Alkuperäisen tietämuskannan suhteiden esittäminen katkoviivalla havainnollistaa sitä, että nämä eivät välttämättä ole suoria, vaan saattavat koostua useasta kaaresta. Kuvassa s osoittaa alika-



Kuva 13: Projektioesimerkki. Kuvassa valkoisella alkuperäisen tietämuskannan resurssit ja kaaret ja harmaalla näistä projisoidut kategoriat ja kohteet.

tegoriasta ylikategoriaan ja t kohteesta tähän liittyvään kategoriaan. Kategoriat viittaavat projektion kohteena olevaan resurssiin kaarella p . Projektion kategoriat ovat aina anonyymejä ja kohteiden URI on sama kuin annotaatioiden. Annotaatiot ovat kuvassa yhdistetty niitä vastaaviin kohteisiin $=$ -kaarella.

Perustavan laatuinen ongelma näkymäpohjaisen RDF-haun toteuttamisessa on se, että RDF-kuvaukset ovat verkkoja siinä missä kategoriahierarkiat ovat luonteeltaan puumaisia. Toiseksi kategorioiden ja kohteiden välinen suhde on näkymäpohjaisen haun kannalta hyvin suoraviivainen. Siihen ei liity muuta semantiikkaa kuin, että kohde liittyy kategoriaan. Sen sijaan tietämuskannassa resurssien ja kohteiden väliset suhteet voivat olla alakohtaisesta ja annotaatio-ontologiasta riippuen mielivaltaisen monimutkaisia. Tässä luvussa käsitellään annotaatio- ja alakohtaisten ontologioiden erojen peittämistä näkymien projisoinnilla. Näkymäprojektiio jakaantuu kategoria- ja kohdeprojektiio-osiin. Kategoriahierarkioiden projisointia käsitellään luvussa 8.1 ja kohteiden projisointia luvussa 8.2.

8.1 Hierarkiaprojektit

Hierarkiasäännöiksi kutsutaan sääntöjoukkoa, jonka perusteella hierarkioiden projisointi suoritetaan. Kategorian suhde yleisiin resurssihin (esimerkiksi luokkiin ja näiden instansseihin) kiteytyy siinä, että kategorialla on yksikäsitteinen polku. Esimerkiksi moniperinnän, moni-instantioinnin ja mahdollisten useiden ristikkäisten näkökulmien (esimerkiksi hyponymia ja meronymia) vuoksi yhtä alakohtaisen tietämuskannan resurssia koh-

Esimerkki 8: Meronyyminen paikkahierarkia.

```

1 :Europe rdf:type :Continent .
2 :Finland rdf:type :Country ;
3   :partOf :Europe .
4 :Helsinki rdf:type :City ;
5   :partOf :Finland .

```

den voi olla n kategoriaa. Erityisesti risteävät näkökulmat aiheuttavat tarpeen tunnistaa, missä roolissa kyseistä resurssia milloinkin tarkastellaan. Esimerkiksi Helsinkiin kaupunkiin ja Helsinkiin Suomessa sijaitsevana paikkana, saattaa liittyä eri kohdejoukot. Edellisen näkökulman kannalta mielekkäitä kohteita ovat esimerkiksi sellaiset museoesineet, joiden käyttäjänä on Helsingin kaupunki. Jälkimmäisen näkökulman kannalta puolestaan mikä tahansa kohde, joka on esimerkiksi valmistettu Helsingin alueella, on kiinnostava.

Yksinkertaisimmillaan hierarkiasääntö koostuu juuresta ja predikaatista, jonka perusteella hierarkia muodostetaan. Jos esimerkiksi paikkahierarkia olisi määritelty esimerkin 8 osoittamalla tavalla, riittäisi tämän hierarkian kuvaamiseen sääntö ”root = :Europe, predicate = :partOf”. Projektio suoritetaan aloittamalla säännön osoittamasta juuresta ja hakemalla siitä lähtien rekursiivisesti hierarkian seuraavat tasot algoritmin PROJECT-1 mukaisesti. Algoritmi huomioi myös syötteen mahdolliset syklit ($C_1 \leftarrow C_2 \leftarrow \dots \leftarrow C_n \leftarrow C_{i \leq n}$) ja refleksiiviset määritelmät ($C_n \leftarrow C_n$). Algoritmin ensimmäinen kutsu on muotoa PROJECT-1($\{\}, root, predicate$).

PROJECT-1(*path*, *node*, *predicate*)

```

1  statements ← findStatements(*, predicate, node)
2  foreach statement in statements
3    subNode ← subjectOfStatement(statement)
4    if subNode not in path and subNode ≠ node
5      process(subNode)    ▷ update data model or visualize
6      PROJECT-1(path ∪ node, subNode, predicate)

```

Funktio *findStatements(subject, predicate, object)* palauttaa annettuja subjektia, predikaattia ja objektia vastaavat lauseet. Jokerina käytetään tähteä (*). Funktio *subjectOfStatement(statement)* palauttaa annetun lauseen subjektin. Varsinainen sovelluskohtainen solmun prosessointi (tietomallin päivitys, visualisointi tai esimerkiksi tulos) suoritetaan funktiolla *process(node)*.

Tämä ratkaisu kattaa esimerkiksi useimmat meronymiatapaukset, joissa osa viittaa kokonaisuuteen *partOf*-predikaatilla tai vastaavalla. Sillä ei kuitenkaan pysty muodostamaan kenties tärkeintä eli hyponymiahierarkiaa. Luokkamallinnuksessa hyponymia jakaantuu kahteen osaan: *supertype* ja *type*. Yksinkertainen ratkaisu olisi sallia yhden predikaatin sijasta usean (*rdfs:subClassOf* ja *rdf:type*) määrittäminen. Algoritmi PROJECT-2 käsittelee tämän kaltaisia määrittäyksiä.

PROJECT-2(*path*, *node*, *predicates*)

```

1  foreach predicate in predicates
2    statements ← findStatements(*, predicate, node)
3    foreach statement in statements
4      subNode ← subjectOfStatement(statement)
5      if subNode not in path and subNode ≠ node
6        process(subNode)    ▷ update data model or visualize
7        PROJECT-2(path ∪ node, subNode, predicates)

```

RDF(S)-pohjaiselle mallinnukselle tämä ei kuitenkaan riitä, sillä myös luokkien kuvaamiseen käytetään samoja predikaatteja kuin instanssien ja näkökulmasta riippuen luokkien ovat instansseja. Vaarana on metatietoarkkitehtuurimallin eri tasojen sekoittuminen keskenään. Esimerkiksi jos OWL-kielellä tehdyn ontologian tarkastelun lähtökohdaksi (juureksi) valittaisiin *rdfs:Class*, muodostuisi hierarkian alkuosa RDF(S)-kielen määrittämisistä, näitä seuraisivat hierarkiassa OWL-kielen määrittäykset, joiden jatkoksi tulisivat varsinaisen ontologian luokat ja lopuksi instanssit. Puurot ja vellit menisivät niin sanotusti sekaisin, sillä informaatiotason ilmentymät eivät ole metamallin saati metametamallin hyponyymejä.

Mahdollistamalla useiden hierarkian muodostavien predikaattien lisäksi näiden luonnehtimisen joko sisäsolmuja tai lehtiä tuottaviksi (*isTransitive* = *false*), onnistuu myös hyponymian esittäminen. Tämän lisäksi predikaatin suunta (käänneinen tai suora, *isInverse*) tulee olla mahdollista määrittellä, sillä vaikka useimmissa tapauksissa osa viittaa kokonaisuuteen (*isInverse* = *true*), samoin kuin hyponyymi hyperonyymiin, on myös toisenlainen mallinnusratkaisu mahdollinen. Esimerkiksi tesauksissa, kuten Yleisessä suomalaisessa asiasanastossa (YSA) [VES00], käytetään termien määrittelyssä sekä *laajempi-termi* että *suppeampi-termi* predikaatteja. Esimerkissä 9 on OWL-määrittely tästä hierarkiasääntöskemasta. Algoritmi PROJECT-3 käsittelee tämän skeeman mukaisia hierarkiasääntöjä.

Esimerkki 9: Hierarkiaprojektioskeema

```

1 :HierarchyDefinition rdf:type owl:Class .
2 :root rdf:type rdf:Property ;
3   rdfs:domain :HierarchyDefinition ;
4   rdfs:range rdfs:Resource .
5 :arcDefinition rdf:type owl:ObjectProperty ;
6   rdfs:domain :HierarchyDefinition ;
7   rdfs:range :ArcDefinition .
8 :ArcDefinition rdf:type owl:Class .
9 :predicate rdf:type rdf:Property ;
10  rdfs:domain :ArcDefinition ;
11  rdfs:range rdf:Property .
12 :isTransitive rdf:type owl:DatatypeProperty ;
13  rdfs:domain :ArcDefinition ;
14  rdfs:range xsd:boolean .
15 :isInverse rdf:type owl:DatatypeProperty ;
16  rdfs:domain :ArcDefinition ;
17  rdfs:range xsd:boolean .

```

PROJECT-3(*hierarchyDefinition*)

```

1 foreach root in roots(hierarchyDefinition)
2   process(subNode)      ▷ update data model or visualize
3   PROJECT-3-SUB1({}, root, arcDefinitions(hierarchyDefinition))

```

```

PROJECT-3-SUB1(path, node, arcDefinitions)
1  foreach arcDefinition in arcDefinitions
2    if isInverse(arcDefinition)
3      statements  $\leftarrow$  findStatements(*, predicate(arcDefinition), node)
4      foreach statement in statements
5        subNode  $\leftarrow$  subjectOfStatement(statement)
6        PROJECT-3-SUB2(path  $\cup$  node, subNode, isTransitive(arcDefinition),
7          arcDefinitions)
8    else
9      statements  $\leftarrow$  findStatements(node, predicate(arcDefinition), *)
10     foreach statement in statements
11       subNode  $\leftarrow$  objectOfStatement(statement)
12       PROJECT-3-SUB2(path  $\cup$  node, subNode, isTransitive(arcDefinition),
13         arcDefinitions)

```

```

PROJECT-3-SUB2(path, node, isTransitive, arcDefinitions)
1  if node not in path
2    process(node)  $\triangleright$  update data model or visualize
3    if isTransitive
4      PROJECT-3-SUB1(path, node, arcDefinitions)

```

Algoritmissa PROJECT-3 *roots*(*hierarchyDefinition*) palauttaa parametrina annetun resurssin *root*-predikaatin arvot ja *arcDefinitions*(*hierarchyDefinition*) *arcDefinition*-predikaatin arvot. Funktiot *predicate*(*arcDefinition*), *isInverse*(*arcDefinition*) ja *isTransitive*(*arcDefinition*) palauttavat vastaavien predikaattien arvot. Funktio *objectOfStatement*(*statement*) palauttaa annetun lauseen objektin.

Mahdolliset epäsuorat yhteydet hierarkiaprojektion kohteena olevien resurssien välillä tulisi tämän skeeman mukaisissa säännöissä muuttaa suoriksi suhteiksi ennen projektiota. Tämä onnistuu tietämuskantaan integroidulla päättelykoneella. Varsinaiset säännöt kohdistuisivat tällöin päätelyihin suhteisiin.

Promoottori luo projektiot Java-ohjelmointikielellä käyttäen Jena [HP 03]. Se ymmärtää vain luokista ja näiden ilmentymistä koostuvia hyponymiahierarkioita. MuseoSuomessa hierarkiaprojektiot kuvataan Prolog-logiikkaohjelmointikielellä. Tämä mahdollistaa mielivaltaisen monimutkaiset säännöt. Oli hierarkiasäännöt kuvattu millä hyvänsä kielellä, tulee niiden käsittelyssä eli projisoinnissa, huomioida ja poistaa myös mahdolliset syklit.

RDF(S)-kielen varhaisissa versioissa oli vaatimus *rdfs:subClassOf*-hierarkioiden syklistömyydestä. Tästä vaatimuksesta ollaan kuitenkin yleiskäyttöisyyden nimissä¹⁶ suosituksen myötä luovuttu.

8.2 Kohdeprojektiot

Hierarkiaprojektion jälkeen tai sen yhteydessä on myös selvitettävä, mitä kohteita kuhunkin kategorian liittyy tai päinvastoin. Tapoihin, joilla kohteet liittyvät kategorioihin, vaikuttavat sekä annotaatio-ontologia että alakohtainen ontologia.

Annotaatio-ontologia vaikuttaa niihin tapoihin, joilla kohteet liittyvät tietämyskannan muihin kohdetta kuvaaviin resursseihin. Käytännössä tämä tapa voi olla myös itse näkymän määrittävä tekijä. Esimerkiksi MuseoSuomessa sama paikkahierarkia toistuu kahdessa eri näkymässä: *valmistuspaikat* ja *käyttöpaikat*. Näiden näkymien erottava tekijä on se, että niiden kategoriat liittyvät haettavaan kohteisiin eri predikaattien (*valmistuspaikka* ja *käyttöpaikka*) kautta. MuseoSuomen hakumoottorissa, Ontogatorissa, tämän kaltaiset erot huomioidaan jo hierarkiaprojektioiden tasolla: muuten samanlaiset hierarkiat käsitellään erillisinä, jos niihin liittyy eri kohteet.

Monimutkaisempien annotaatio-ontologioiden suhteen ei yksittäisten kohteet kategorioihin liittävien predikaattien määrittely riitä. Yksi looginen annotaatio voi koostua useiden luokkien instansseista ja niiden välisistä suhteista. Esimerkiksi artikkelissa *Ontology-based Photo Annotation [SDWW01]* kuvattu annotaatio-ontologia käsittelee annotaatiota kolmesta näkökulmasta:

Media Mediapiirteitä (Medium feature) ovat esimerkiksi kuvan resoluutio ja formaatti.

Valokuva Valokuvapiirteitä (Photo feature) ovat esimerkiksi kuvaaja, tarkka aika ja paikka.

Aihe Aiheen kuvaukseen (Subject-matter description) liittyvät muun muassa kuvassa näkyvät objektit, tapahtumat, relatiivinen aika ja paikka.

Nämä näkökulmat mallinnetaan erillisinä luokkina, jotka liittyvät varsinaiseen annotaatioon (Photo annotation). Lisäksi aihe kuvataan rakenteisesti tavalla, joka muistuttaa luonnollista kieltä. Aihekuvaus koostuu neljästä elementistä:

¹⁶Syklistä perintää voidaan käyttää ekvivalenssin ilmaisemiseen.

1. *Toimija* (esim. ”apina”), jolla voi edelleen olla kuvaavia piirteitä (modifiers), kuten ikä tai väri (esim. ”oranssi”).
2. *Toiminto*, kuten ”syöminen”.
3. *Objekti* eli toiminnan kohde (esim. ”banaani”), jolla voi myös olla kuvaavia piirteitä (esim. ”vihreä”).
4. *Miljöö*, joka kuvaa esimerkiksi tapahtuman relatiivisen ajan ja paikan (esim. ”metssä auringonlaskun aikaan”).

Alakohtainen ontologia puolestaan vaikuttaa siihen, mitkä kaikki kyseiseen hierarkiaan kuulumattomat resurssit ovat relevantteja annotaation arvona olevan resurssin suhteen. Siinä, missä yksi looginen annotaatio voi olla rakenteinen kuvaus, voivat myös alakohtaisen ontologian käsitteet olla rakenteisia. Tällaiset resurssit ovat tyypillisesti korkeamman tason käsitteitä, joiden kuvaamiseen ei riitä yksi resurssi tai binäärinen suhde. Esimerkiksi promootiokuvissa henkilö voi esiintyä kuvassa joko omana itsenään tai jossain roolissa, kuten rehtorina. Järjestelmän on kyettävä löytämään Kari Raiviota haettaessa myös sellaiset kuvat, joissa hän on rehtorina. Tapauksesta riippuen myös jotkin funktionaaliset suhteet saattavat olla assosiatiivisia. Esimerkiksi promootio-tapahtumilla on järjestäjänä aina jokin promootiotiedekunta. Promootiotiedekunta puolestaan saattaa koostua useista todellisista tiedekunnista, kuten historiallinen Filosofinen tiedekunta koostuu nykyisistä Matemaattis-luonnontieteellisestä, Humanistisesta ja Kasvatustieteellisestä tiedekunnasta.

Näkymäpohjaisen haun kannalta tämänkaltaisia monimutkaisia suhteita ei tarvitse ymmärtää. Riittää, että hakukone ymmärtää kohteen liittyvän johonkin kategoriaan. Sääntöjä, joiden perusteella kategorioiden liittyvät kohteet projisoidaan kutsutaan kohdeprojektiosäännöiksi (mapping rules). *Kohdeprojektiosäännöt* määrittävät ne tavat, joilla kohteet liittyvät kategorioihin. Ne vastaavat siis kysymykseen ”mitä kohteita tähän (kategoriaprojektion kohteena olevaan) resursiin liittyy” [HJK⁺04].

MuseoSuomi-järjestelmässä kohdeprojektiosäännöt määritellään Prologilla. Promootorissa kohdeprojektiosäännöt ilmaistaan RDF-polkuina. Ne annetaan aina jollekin tietylle luokalle ja periytyvät tämä aliluokille ja instansseille. Polkujen syntaksi vastaa N3-kielen polkumäärittäjiä [BLSC03]. RDF-polut koostuvat predikaateista, joiden edessä oleva piste (.) tai hattu (^) kertoo kaaren suunnan. Esimerkiksi Henkilöt-luokalle annetut kohdeprojektiosäännöt

1. `^promootio:roolissa_olevat_henkilöt^promootio:mihin_instansseihin_liittyy`

2. \wedge *promootio:mihin_instansseihin_liittyy*

kertovat, että Henkilö-luokka¹⁷, sen aliluokat ja instanssit liittyvät kohteisiin (kuvakortteihin) joko siten, että 1) kuvakortista on suora *promootio:mihin_instansseihin_liittyytyyppinen* viittaus kyseiseen resurssiin tai sitten 2) tämä edelleen viittaa johonkin, jonka *promootio:roolissa_olevat_henkilöt_kyseinen* resurssi on.

9 Ontogator

Ontogator on näkymäpohjainen RDF-hakumoottori, joka toteuttaa näkymäpohjaisen hakukoneen hakulogiikan. Sen avulla on mahdollista toteuttaa erilaisia näkymäpohjaisia hakukoneita. Ontogatorin RDF-pohjaisen kyselyrajapinnan suunnittelussa on huomioitu HiBrowse-, Promootori-, Flamenco- ja MuseoSuomi-tyyppiset näkymäpohjaiset hakukoneet.

Ontogatorin rajapinta on määritelty OWL-kielellä ja se mahdollistaa kategorioiden ja kohteiden hakemisen sekä erikseen että yhdessä siten, että kohteet ovat ryhmitelty kategoriottain. Kategorioita ja kohteita on mahdollista hakea sekä tunnisteella että avainsanapohjaisesti. Lisäksi kategorioita on mahdollista hakea projektion kohteena olevan resurssin ja kohteita niihin liittyvien kategorioiden perusteella. Annetun kohteiden hakuehdon perusteella Ontogator suodattaa vastauksesta sellaiset kategoriat, joihin ei liity kohteita. Vaihtoehtoisesti suodatuksen (esimerkiksi harmaannuttamalla) voi jättää käyttöliittymän vastuulle luvussa 7.2 kuvatulla tavalla.

Sekä kohteisiin että kategorioihin on mahdollista liittää sekä dynaamista että staattista metadataa. Dynaamista metadataa ovat esimerkiksi kategorioihin annetuilla hakuehdoilla liittyvien kohteiden ja suodattamattomien alikategorioiden määrät. Staattista metadataa ovat esimerkiksi kohteiden ominaisuudet kuten kuvatiedoston nimi ja kuvausteksti. Kohteisiin on myös mahdollista liittää niitä kuvaavat kategoriat polkuineen.

Ontogatoria on kehitetty Helsingin yliopiston tietojenkäsittelytieteen laitoksen Semantic Computing -tutkimusryhmässä vuoden 2002 alusta lähtien. Tänä aikana Ontogatorista on kehitetty useita eri versioita. Ensimmäinen prototyyppi kehitettiin opiskelijatyönä WWWMuseo-nimisessä ohjelmistotuotantoprojektissa promootio-aiheisten kuvien WWW-näyttelyksi. Toinen versio, Promootori, on Javalla toteutettu kioskisovellus, joka on tällä hetkellä tuotantokäytössä Helsingin yliopistomuseossa. Promootorin aineisto-

¹⁷On sinänsä oliomallinnustavan vastaista, että viittaus voi kohdistua myös luokkamääritykseen. Promootio-ontologian mallinnusratkaisujen vuoksi tämä on kuitenkin välttämätöntä.

na on promootio-aiheiset kuvat, jotka on annotoitu Avril Styrmanin tekemän Promootio-ontologian mukaisesti. Näissä Ontogatorin kehitysversioissa hakulogiikka oli osa itse järjestelmää. Ontogatorin uusin versio erottaa hakulogiikan täysin käyttöliittymästä, annotaatio-ontologiasta ja alakohtaisesta ontologiasta. Sitä käytetään MuseoSuomi-järjestelmän hakumoottorina. Lisäksi sen avulla on rakennettu palveluiden hakemiseen tarkoitettun hakukoneen prototyyppi *Intelligent Web Services* -projektissa¹⁸. Tässä tutkielmassa Ontogatorilla viitataan nimenomaan tähän versioon, jonka kehitys aloitettiin vuoden 2003 ke-sällä.

Ontogatorin syötteessä, kyselyissä ja vastauksissa käytetty Ontogator-ontologia on liitteenä (liite B). Ontologian nimiavaruus, johon tässä tutkielmassa viitataan etuliitteellä *ogt*, on <http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#>. Se laajentaa W3C:n Bookmark-ontologiaa (liite A). Bookmark-ontologian nimiavaruus, johon viitataan etuliitteellä *bm*, on <http://www.w3.org/2002/01/bookmark#>. Kuvassa 14 on Ontogatorin syöte-, vastaus- ja kyselyskeeman eli Ontogator-ontologian UML-kuvaus sekä Bookmark-ontologian keskeisimmät luokat ja ominaisuudet.

Ontogatorin julkinen rajapinta jakaantuu kolmeen osaan: syöte-, kysely- ja vastausraja-pintaan. Luvussa 9.1 käsitellään syöterajapintaa, luvussa 9.2 kyselyrajapintaa ja luvussa 9.3 vastausrajapintaa. Luvussa 9.4 esitellään implementaatiota.

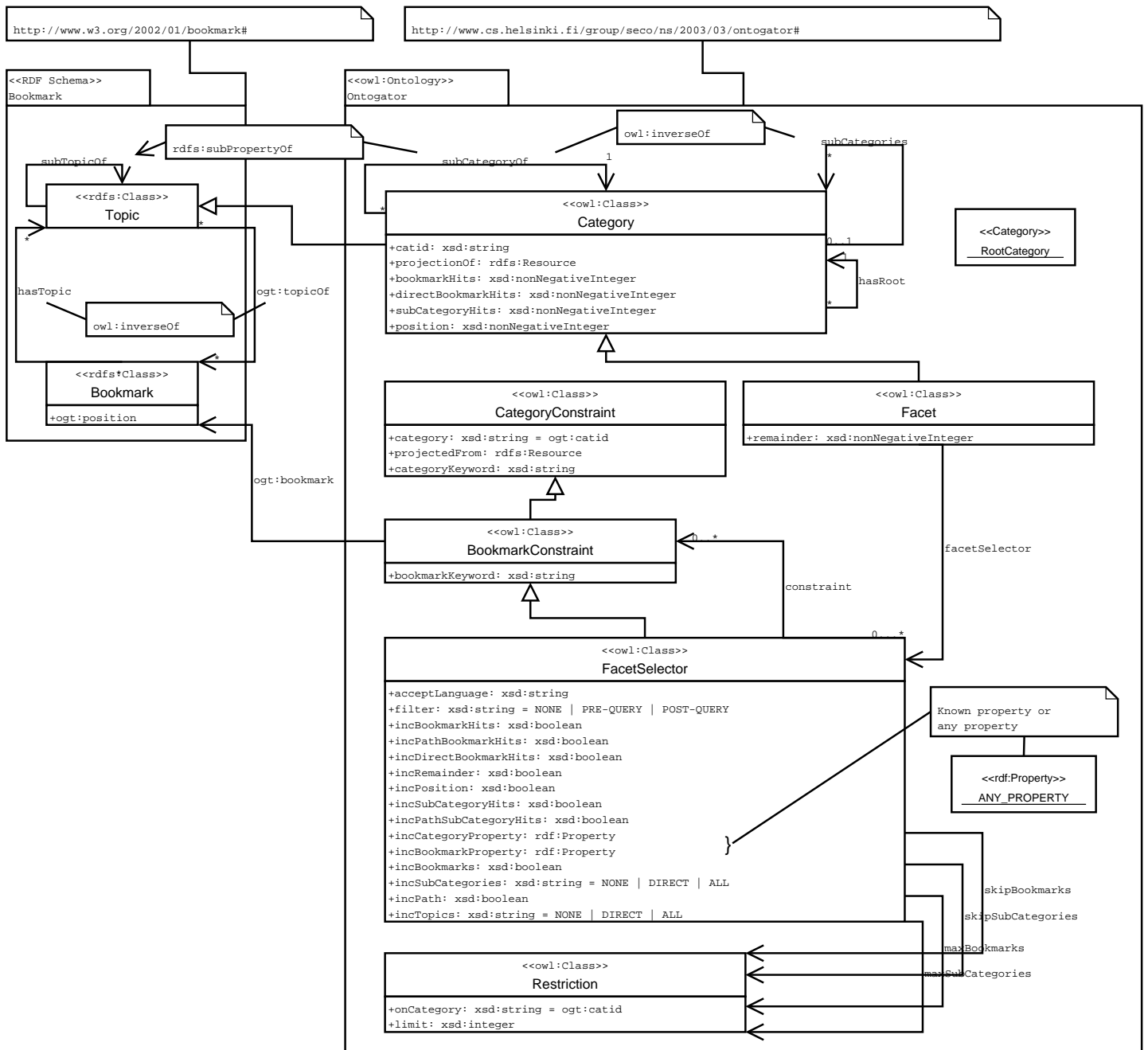
9.1 Syöterajapinta

Tässä aliluvussa käsitellään Ontogatorin projektiosyöterajapintaa. Projektion tietomalli vastaa luvussa 8 kuvattua ja se koostuu kategorioista ja kohteista sekä näiden ominaisuuksista ja välisistä suhteista (ks. kuva 13). Projektion avulla hakumoottori on mahdollista toteuttaa aineistoriippumattomaksi niin, ettei sen tarvitse ymmärtää kohteiden kuvauksessa käytettyjä annotaatio ja alakohtaisia ontologioita. Erilaiset aineistot on näin mahdollista saattaa saman hakupalvelun piiriin.

Ontogatorin kategoriat ovat *ogt:Category*-luokan ilmentymiä (kuva 14). Tämä perii Bookmark-ontologian luokan *bm:Topic*. Kategoriat viittaavat projektion kohteena olevaan resurssiin *ogt:projectionOf*-predikaatilla (kuvassa 13 *p*-kaaret) ja niillä voi olla useita erikielisiä nimiöitä (*rdfs:label*).

Ogt:Category-luokan olennaisin ero verrattuna *bm:Topic*-luokkaan on se, että kategoriahierarkiat ovat yksiperintäisiä siinä missä aihehierarkiat ovat moniperintäisiä. Tämä rajoitus johtuu siitä, että käyttöliittymässä on tiedettävä minkä nimenomaisen solmun

¹⁸<http://www.cs.helsinki.fi/group/iwebs/>



Kuva 14: Ontogatorin syötte-, vastaus- ja kyselyskeeman eli Ontogator-ontologian (liite B) UML-kuvaus.

käyttäjä on valinnut. Promoottorin varhaisessa kehitysversiossa solmut tunnistettiin URI-tunnisteen perustella moniperintäisessä luokkahierarkiassa. Tämä aiheutti sen, että valittaessa yksi moniperintäinen solmu, useita solmuja korostui. Ongelmallista tämä oli ennen kaikkea sen vuoksi, että Promootio-ontologian rakenteen vuoksi sama solmu saattoi olla samassa hierarkiassa sekä lehtenä (instanssina) että sisäsolmuna (luokkana). Näin ollen kyseisiin solmuihin liittyi eri kohteet, eikä käyttöliittymän perusteella pystynyt päätellä kumpi solmuista oli todellisuudessa valittu.

Myös `bm:Topic`- ja `ogt:Category`-luokan käyttötavat poikkeavat olennaisesti toisistaan: kategoriat ovat loogisesti projektioita, eivät niinkään itsenäisiä resursseja. Monet kategorioiden ominaisuuksista määräytyvät dynaamisesti hakuehdon perusteella. Tämä puolestaan vaikuttaa muun muuassa siihen, että kategorioiden tulee olla anonyymejä resursseja. Näin vältetään kahden eri kyselyyn vastaavan kategorian tietojen sekoittuminen keskenään, jos vastausta tulkitaan RDF-muotoisena.

Kategoriahierarkiat määritellään predikaatilla `ogt:subCategories` listana alikategorioita. Kaikkien kategoriahierarkioiden yhteinen juuri on `ogt:RootCategory`. Kuvassa 13 alikategoriat määritellään kaarella `s` siten, että alikategoria viittaa ylikategoriaan. Ontogatorin projektiosyötteessä käytetään tälle käänteistä listamuotoista esitystapaa, jotta kohteiden välinen järjestys voitaisiin taata. Paitsi käyttöliittymään, kohteiden keskinäinen järjestys vaikuttaa myös kategorioiden tunnisteisiin. Listamuotoinen esitystapa on välttämätön, jotta syötteen pysyessä muuttumattomana myös kategorioille projektion (uudelleen)loutauksen yhteydessä luotavat tunnisteet pysyisivät samoina.

Kuhunkin kategoriaan suoraan liittyvät kohteet määritellään predikaatilla `ogt:topicOf`. Sen arvona on lista (`rdf:List`) `bm:Bookmark`-tyyppisiä resursseja. Kuvassa 13 on kategorioihin liittyvät kohteet esitetty kaarella `t` siten, että kohteet viittaavat kategorioihin. Ontogatorin projektiosyötteessä käytetty käänteinen listamuotoinen viittaustapa johtuu siitä, että vain täten voidaan taata kohteiden keskinen järjestys.

Projisointivaiheessa kategorioihin voidaan moni-instantioinnin kautta liittää myös muita kuin `ogt:Category`-luokan ominaisuuksia. MuseoSuomessa kategorioihin liitetään suoraan kaikki projektion kohteen ominaisuudet. Näitä voidaan käyttää käyttöliittymässä esimerkiksi antamaan lisätietoja kategorioista. Myös hakuja olisi periaatteessa helppoa kohdistaa mihin tahansa näistä ominaisuuksista. Tätä ei ole kuitenkaan toistaiseksi toteutettu. Nykyrajapinta mahdollistaa vain projektion kohteen perusteella hakemisen sekä sanahaussa käytettävien ominaisuuksien konfiguroinnin.

9.2 Kyselyrajapinta

Tässä aliluvussa käsitellään Ontogatorin kyselyrajapintaa. Kyselyontologia on pyritty rakentamaan siten, että se mahdollistaa hyvin monenlaisten näkymäpohjaisten hakusovellusten toteutuksen ja optimoimisen. Kyselyjä on mahdollista optimoida erilaisilla rajoitteilla ja optioilla. Optimointien peruseriaate on, että Ontogatorilta pyydetään vain sellaiset tiedot, joita todella tarvitaan.

Kyselyt koostuvat kolmesta osasta. Luvussa 9.2.1 käsitellään kategoriaehtoja, luvussa 9.2.2 kohde-ehtoja ja luvussa 9.2.3 kyselyiden varsinaisia selektoreita eli näkymäehtoja.

9.2.1 Kategoriaehtot

Kategoriaehtoja käytetään valitsemaan kategorioita. Perustapauksessa kategoriaan viitataan sen tunnisteella, mutta kategorioita voi hakea myös avainsanan tai projektion kohderessin perusteella. Kuvassa 9 kategorioita on haettu sekä kategoriatunnisteen että hakusanan perusteella. Projektion perusteella kategorioita haetaan esimerkiksi Museo-Suomen mobiilikäyttöliittymässä käyttäjän sijaintiin perustuvissa hauissa. Niissä erillinen paikannuspalvelin palauttaa joukon ontologisesti määriteltäviä paikkaresursseja, joita edelleen käytetään projektiohakuena Ontogatorille.

Kategoriaehtoa käytetään riippuen näkymäehdon optioista (ks. luku 9.2.4) ja roolista suhteessa tähän

1. *kategorioiden hakemiseen* osana näkymäehtoa, jos vastaukseen halutaan kategoriat, mutta ei kohteita,
2. *ryhmittelyehtona* osana näkymäehtoa, jos vastaukseen halutaan sekä kategoriat että kohteet, ja/tai
3. *kohteiden hakemiseen* kohde-ehtona, osana näkymäehdon kohteiden rajoitusehtoa (ks. luku 9.2.2).

Kategoriaehdot ovat luokan `ogt:CategoryConstraint` ilmentymiä (kuva 14). Varsinainen ehto annetaan kolmella predikaatilla, jotka ovat suoritusjärjestyksessä:

1. *ogt:category* —rajoittaa kategorioita *ogt:catid* suhteen,

2. *ogt:projectedFrom* —rajoittaa kategorioita *ogt:projectionOf* suhteen (projektioehto) ja
3. *ogt:categoryKeyword* —rajoittaa kategorioita konfiguroitavissa olevista literaaliarvoisista predikaateista (esimerkiksi *rdfs:label*) löytyvän osamerkkijonon suhteen (sanahakuehto).

Jos yhtään kategoriarajoitetta ei anneta, vastaa kategoriaehto juurikategorioihin. Mitä tahansa näistä rajoitteista voi antaa 0–n kappaletta. Jos *ogt:category*-ehtoja on useita, saadaan tulokseksi yhdiste näistä kategorioista. Vastaavasti, jos *ogt:projectedFrom* tai *ogt:categoryKeyword* ehtoja on useita, saadaan tulokseksi yhdiste näitä vastaavista kategorioista.

Jos kysely käsittää yhdistelmiä eri tyyppisistä kategoriarajoitteista, kohdistuvat vähemmän rajoittavat ehdot aina enemmän rajoittavien ehtojen määrittämiin kategoriahierarkioihin yllä määritellyssä suoritusjärjestyksessä. Näin on mahdollista kohdistaa projektioehto ja/tai sanahakuehto määrättyyn hierarkiaan tai hierarkioihin; hakea esimerkiksi merkkijonoa ”Espoo” vain *Käyttöpaikka*-näkökulmasta. MuseoSuomessa tätä mahdollisuutta ei toistaiseksi hyödynnetä.

9.2.2 Kohde-ehdot

Kohde-ehdoja käytetään valitsemaan kohteita osana näkymäehtojen kohteiden rajoitusehtoa (ks. luku 9.2.3). Kohteiden rajoitusehto voi koostua useista kohde-ehdoista ja tuloksena saadaan näiden leikkaus eli kohteet, jotka toteuttavat kaikki annetut ehdot. Kohde-ehdo koostuu kategorია-, kohde- ja avainsanarajoitteista.

Kohde-ehdon määrittelevä luokka (*ogt:BookmarkConstraint*) on kategoriaehtojen (*ogt:CategoryConstraint*) aliluokka. Kohde-ehdon kategoriarajoite määritellään luvussa 9.2.1 esitetyllä tavalla. Kohde-ehdo voi käsittää vain yhden loogisen kategoriarajoitteen.

Kategoriarajoitteiden ohella kohde-ehdo voi koostua kohde- ja avainsanarajoitteista. Kohderajoite (*ogt:bookmark*) rajoittaa kohteita URI-tunnisteen perusteella. Sitä käytetään esimerkiksi MuseoSuomen kohdesivun (kuva 12) tuottavassa kyselyssä hakemaan valitun kohteen tiedot. Avainsanarajoite (*ogt:bookmarkKeyword*) rajoittaa kohteita konfiguroitavissa olevista literaaliarvoisista predikaateista (esimerkiksi *rdfs:label* ja kohteen kuvausteksti) löytyvän osamerkkijonon suhteen (sanahakuehto). Tätä käytetään MuseoSuomen sanahaussa yhdessä kategorioiden sanahaun kanssa (kuva 9). Kohde-ehdo voi käsittää useita kohde- ja avainsanarajoitteita. Kohde-ehdo tuottaa yhdisteen kaikkien siihen

liittyvien rajoitteiden löytämistä kohteista. Jos yhtään rajoitetta ei ole määritelty, vastaa kohde-ehto kaikkia järjestelmän tuntemia kohteita.

9.2.3 Näkymäehdot

Näkymäehto on varsinainen kysely, joka voi käsittää joukon *optioita*, *rajoitteita*, *kategoriahakuehdon* sekä *kohteiden rajoitusehdon*. Yksi kyselydokumentti voi sisältää useita mahdollisesti toisistaan riippumattomia näkymäehtoja. Tyypillinen kysely käsittää vähintään yhden näkymäehdon kutakin näkymää kohden. Varsinaisten näkymien lisäksi näkymäehtoja käytetään esimerkiksi MuseoSuomen hakusivulla (kuva 9) kategoriahakuehdojen perustietojen (polku ja nimiö) hakemiseen (kuvassa keskellä ylhäällä) sekä kohteiden hakemiseen. Hakuehdojen perustiedot haetaan erillisellä kyselyllä koska esimerkiksi sanahaku voi johtaa tilanteeseen, jossa nämäkin kategoriat suodattuisivat pois. Myös kohdesivulla näytettävät tiedot (oikeassa laidassa näkyviä suosituksia lukuun ottamatta) haetaan Ontogatorilta käyttäen näkymäehtoa (kuva 12).

Kuvan 15 esimerkissä on RDF/XML-muodossa kolme näkymäehtoa (*objectType*, *espoo* ja *bookmarksByCategory*), joista kahta (*objectType* ja *espoo*) käytetään kohteiden rajoitusehtona kaikille kolmelle. Kysely vastaa optioineen ja rajoitteineen osaa MuseoSuomen hakusivun (kuva 9) tuottavasta kyselystä. Näkymäehto *objectType* on peruskategoriakysely ja *espoo* sanahakuehto. Kolmatta näkymäehtoa (*bookmarksByCategory*) käytetään kohteiden hakemiseen. Esimerkissä näkymäehtoja *objectType* ja *espoo* käytetään yksinomaan kategorioiden hakemiseen. Näkymäehtoa *bookmarksByCategory* käytetään kategorian *säilytysastiat* mukaisesti ryhmiteltyjen kohteiden hakemiseen. Esimerkkiä on muokattu esimerkiksi URI-tunnisteiden osalta alkuperäistä luettavampaan muotoon. Alkuperäinen kysely on kokonaisuudessaan liitteenä (liite C) samoin Ontogatorin vastaus tähän kyselyyn (liite D).

Ontogator vastaa kuhunkin näkymäehtoon fasetilla (*ogt:Facet*-luokan ilmentymä). Jos näkymäehdolla on URI-tunniste, viittaa fasetti siihen predikaatilla *ogt:facetSelector*. Ontogator ei takaa, että kyselyt käsitellään kyselydokumentin määrittämässä järjestyksessä joten, jos kyselyssä on useita anonyymeja näkymäehtoja, ei vastauksesta pysty varmuudella erottamaan mikä fasetti vastaa mihinkin näkymäehtoon.

Periaatteessa kaikki näkymäehdon osat ovat optionaalisia ja yksinkertaisin mahdollinen kysely onkin lauseke, joka määrittelee anonyymien näkymäehdon: ”[*rdf:type ogt:FacetSelector*]”. Tämä kysely ei määrittele rajoitteita. Optioina käytetään oletusarvoja. Vastaukseksi tällaiseen kyselyyn saadaan pelkät juurikategoriat, niiden nimiöt (*rdfs:label*)

```

<ogt:FacetSelector rdf:ID="objectType" ▷ Näkymäehto objectType (Esinetyyppi).
  ogt:incCategories="true" ▷ Haetaan kategorioita (subCategories),
  ogt:incBookmarks="false" ▷ ...mutta ei kohteita (topicOf).
  ogt:incSubCategories="DIRECT" ▷ Haetun kategorian suorat alikategoriat mukaan.
  ogt:incSubCategoryHits="true" ▷ Hakuehtoihin sopivien kategorioiden lukumäärät (subCategoryHits).
  ogt:incBookmarkHits="true" ▷ Kohteiden lukumäärät (bookmarkHits).
  ogt:acceptLanguage="fi" ▷ Literaalit suomeksi.
  ogt:filter="POST-QUERY" ▷ Kategorioiden suodatus kohteiden rajoitusehdon perusteella.
  ogt:incPath="true" ▷ Haetun kategorian polku mukaan.
  <ogt:incCategoryProperty rdf:resource="&fms;rootcatid"/> ▷ Kategorioihin mukaan fms:rootcatid, jos tiedossa.
  <ogt:category>%00%0C%0F</ogt:category> ▷ Haetun kategorian (säilytysasiat) tunniste on %00%0C%0F.
  <ogt:maxSubCategories ogt:limit="20"/> ▷ Vastaukseen korkeintaan kaksikymmentä kategoriää kullekin tasolle.
  <ogt:constraint rdf:parseType="Collection"> ▷ Kohteiden rajoitusehto koostuu kahdesta kohde-ehdosta.
    <rdf:Description rdf:about="#objectType"/>
    <rdf:Description rdf:about="#espo0"/>
  </ogt:constraint>
</ogt:FacetSelector>

<ogt:FacetSelector rdf:ID="espo0" ▷ Hakusanan mukaisesti nimetty näkymäehto espo0.
  ogt:incSubCategoryHits="true"
  ogt:incSubCategories="NONE"
  ogt:incCategories="true"
  ogt:incBookmarks="false"
  ogt:acceptLanguage="fi"
  ogt:filter="POST-QUERY"
  rdfs:label="espo0"
  ogt:incPath="true"
  <ogt:incCategoryProperty rdf:resource="&fms;rootcatid"/>
  <ogt:categoryKeyword>espo0</ogt:categoryKeyword> ▷ Haetaan kategorioita avainsanan "espo0" perusteella.
  <ogt:bookmarkKeyword>espo0</ogt:bookmarkKeyword> ▷ Haetaan kohteita avainsanan "espo0" perusteella.
  <ogt:maxSubCategories ogt:limit="20"/>
  <ogt:constraint rdf:parseType="Collection">
    <rdf:Description rdf:about="#objectType"/>
    <rdf:Description rdf:about="#espo0"/>
  </ogt:constraint>
</ogt:FacetSelector>

<ogt:FacetSelector rdf:ID="bookmarksByCategory" ▷ Näkymäehto bookmarksByCategory.
  ogt:incBookmarks="true" ▷ Haetaan kohteita...
  ogt:incCategories="true" ▷ ...ryhmiteltyinä kategoriottain.
  ogt:incSubCategories="DIRECT" ▷ Kohteet ryhmiteltyinä haetun kategorian ja sen suorien alikategorioiden mukaisesti.
  ogt:filter="POST-QUERY"
  ogt:incPath="true"
  ogt:incTopics="NONE" ▷ Kohteisiin ei liitetä tietoa niihin liittyvistä kategorioista.
  ogt:incDirectBookmarkHits="true" ▷ Vastauksen sisäkategorioihin tieto niihin suoraan liittyvien kohteiden määristä.
  ogt:incRemainder="true" ▷ Tieto ryhmittelyyn sopimattomien kohteiden määrästä mukaan.
  ogt:acceptLanguage="fi"
  ogt:incSubCategoryHits="true"
  <ogt:incCategoryProperty rdf:resource="&fms;rootcatid"/>
  <ogt:incBookmarkProperty rdf:resource="&fms;www_kuvatiedostonimi"/>
  <ogt:incBookmarkProperty rdf:resource="&fms;www_numero"/> ▷ Kohteisiin mukaan fms:www_numero, jos tiedossa.
  <ogt:category>%00%0C%0F</ogt:category> ▷ Ryhmittely kategorian %00%0C%0F (säilytysasiat) perusteella.
  <ogt:maxBookmarks ogt:limit="4"/> ▷ Kohteita maksimissaan neljä kustakin ryhmästä.
  <ogt:maxSubCategories ogt:limit="7"/> ▷ Kategorioita maksimissaan seitsemän.
  <ogt:constraint rdf:parseType="Collection">
    <rdf:Description rdf:about="#objectType"/>
    <rdf:Description rdf:about="#espo0"/>
  </ogt:constraint>
</ogt:FacetSelector>

```

Kuva 15: Tavanomainen kolmen näkymäehdon Ontogator-kysely: Näkymäehtoja *objectType* ja *espo0* käytetään rajoitusehtoina kaikissa kolmessa näkymäehdossa. Näkymäehto *objectType* on peruskategoriakysely ja *espo0* sanahakuehto. *bookmarksByCategory*-ehtoa käytetään hakemaan kohteet ryhmiteltyinä kategorian *säilytysasiat* ja sen suorien alikategorioiden mukaisesti. Kyselyn kommentteissa viitataan vastauksen predikaatteihin (ks. kuva 17 ja liite D). Hakutulokset on visualisoitu kuvassa 9.

ja kategorioiden tunnisteet (*ogt:catid*) sekä tiedot niihin liittyvien kohteiden määristä. Oletuskyselyyn ei tule kohteita mukaan, jotta kyselyn suoritus aika ja vastauksen koko pysyisivät mielekkäissä rajoissa.

Tietämyskannan metatiedot ilmaistaan vastausdokumenttiin viittaavalla (*rdf:about=""*) *ogt:CategoryBase*-luokan ilmentymällä (ks. liite D). Tietämyskannan metatietoa ovat esimerkiksi datan ja käytetyn projektiosäännösten versiot. Koska kategorioiden tunnisteet riippuvat hierarkian rakenteesta, saattavat ne muuttua aina, kun syöte tai projektiosäännöt muuttuvat. Metatiedon avulla käyttöliittymän on mahdollista tunnistaa virheelliset viittaukset.

Näkymäehtoon liittyvien *kohteiden rajoitusehto* annetaan predikaatilla *ogt:constraint* listana kohde-ehtoja (ks. luku 9.2.2). Koska luokka *ogt:FacetSelector* perii luokan *ogt:BookmarkConstraint*, voidaan näkymäehtoja käyttää myös kohde-ehtoina. Yksittäisten kohde-ehtojen löytämistä kohteista tehdään leikkaus. Jos kohteiden rajoitusehtoa ei ole määritetty, valitsee näkymäehto kaikki järjestelmän tuntemat kohteet. Jos esimerkiksi kategoriaehto on tarkoitettu samalla kohteidenhakuehdoksi, ei vain ryhmittelyehdoksi tai kategoriahauksi, tulee näkymäehdon viitata itseensä kohteiden rajoitusehdossa. Kuvassa 15 näkymäehdot *objectType* ja *espool* viittaavat paitsi toisiinsa, myös itseensä. Näkymäehto *bookmarksByCategory* viittaa kohteiden rajoitusehdossa vain näihin kahteen muuhun näkymäehtoon.

Näkymäehdon varsinaisesti tuottamat tiedot määritellään käyttäen optioita ja rajoitteita. Ontogator-pohjaisten järjestelmien optimointi perustuu siihen, että vastaukseen otetaan vain ne tiedot, joita todella tarvitaan. Optioita käsitellään luvussa 9.2.4 ja rajoitteita luvussa 9.2.5.

9.2.4 Optiot

Vastaukseen mukaan otettavat tiedot määritellään kyselyssä optioilla. Tässä luvussa kuvataan tärkeimmät Ontogatorin kyselyrajapinnan optiot yleisellä tasolla.

Totuusarvoinen optio *ogt:incCategories* määrittää, halutaanko vastaukseen kategorioita. Esimerkiksi kohdesivun (kuva 12) tuottava kysely hakee ensisijaisesti kohteita, ei kategorioita. Vastaavasti *ogt:incBookmarks* määrittää, halutaanko vastaukseen kohteita. Jos vastaukseen halutaan sekä kategorioita että kohteita, pyritään kohteet ensisijaisesti sijoittamaan haettavaan kategoriahierarkiaan mahdollisimman syväälle. Sellaiset kohteet, jotka eivät löydy kategoriahierarkiasta liitetään suoraan fasettiin. Tällaiseen tilanteeseen joudutaan esimerkiksi, jos hakuehto käsittää kohdehakusanoja (*ogt:bookmarkKeyword*) tai jos

ryhmittelyehto ei ole osa hakuehtoa.

Kuhunkin kategoriaan (myös fasettiin) liittyvien erillisten (distinct) kohteiden lukumäärät saa vastaukseen totuusarvoisella optiolla *ogt:incBookmarkHits* ja kategorioiden määrät optiolla *ogt:incSubCategoryHits*. Osumien lukumäärien laskemisessa huomioidaan, että sama kohde voi toistua saman hierarkian useissa eri haaroissa. Fasettiin suoraan liittyvien, mahdolliseen ryhmittelyyn sopimattomien kohteiden lukumäärän saa vastaukseen optiolla *ogt:incRemainder*.

Kategoriaehdon löytämät kategoriat esitetään fasetin suorina alikategorioina. Vastaukseen otettavan kategoriahierarkian syvyys annetaan predikaatilla *ogt:incSubCategories*. Vaihtoehdot ovat

NONE vain suorat kategoriaehdon määrittämät kategoriat,

DIRECT kategoriaehdon määrittämän kategoriat sekä näiden suorat alikategoriat ja

ALL kategoriaehdon määrittämän kategoriat sekä näiden kaikki alikategoriat.

Vaihtoehtoa *DIRECT* käytetään MuseoSuomen kategorianäkymissä ja kategorian perusteella ryhmittelyjen kohteiden hakemisessa (kuva 9). Sanahaun vastaukseen otetaan vain suorat (*NONE*) osumat (kuva 9) ja koko luokittelun näyttävällä sivulla käytetään vaihtoehtoa *ALL* (kuva 10).

MuseoSuomen kohdesivulla (kuva 12) näytetään kohteeseen liittyvät kategoriat, jotka saadaan vastaukseen optiolla *ogt:incTopics*. Vaihtoehdot ovat

NONE kohteisiin ei liitetä kategoriatietoja,

DIRECT vain kohteisiin liittyvät suorat kategoriat ja

ALL kohteisiin liittyvät kategoriat polkuineen.

Toivottu suodatustapa (ks. luku 7.2) määritellään näkymäehdon predikaatilla *ogt:filter*. Vaihtoehdot ovat

NONE Ei suodatusta: tyhjään hakutulokseen johtavien kategorioiden asianmukainen visualisointi jää käyttöliittymän vastuulle.

PRE-QUERY Suodatus ennen hakuehtoja: suodatetaan sellaiset kategoriat, joihin ei alunperinkään liity yhtään kohdetta. Tätä suodatustapaa käytetään MuseoSuomen ”näytä kaikki kategoriat”-hakumoodissa (kuvat 11 ja 16).

The screenshot shows the MuseoSuomi website search results. The search criteria are ' espoo ' and 'säilytysasiat'. The results are categorized by 'säilytysasiat, kohteet 1-2/2 (näytä kaikki)'. The first two results are 'lampu-umpikupu (ECM 2037 37)' and 'lampu-umpikupu (ECM 2978 3)'. The next category is 'kanisterit, kohteet 1-4/7 (ryhmittele kohteet)', showing four 'kanisteritaskumatti (ECM)' items with IDs 2194 66, 2199 168, 2199 169, and 2423 253. The final category is 'maljakot, kohteet 1-4/11 (ryhmittele kohteet)', showing four 'maljakot' items with various designs.

Kuva 16: MuseoSuomen hakusivu. Hakuehtoina hakusana ” espoo ” ja kategoria *säilytysasiat*. Suodatetut kategoriat harmaannutettuina

POST-QUERY Suodatus hakuehtojen jälkeen: suodatetaan annettujen hakuehtojen perusteella tyhjiin tulokseen johtavat kategoriat. Tämä on oletusarvoinen suodatustapa ja sitä käytetään MuseoSuomessa aina kohteiden ryhmittelyssä.

Kategorioihin projisointivaiheessa liitettyjä ominaisuuksia voi tiedustella optiolla *ogt:incCategoryProperty*, jonka määrittämä predikaatti, mikäli tiedossa, liitetään kategorioihin. Tätä käytetään MuseoSuomessa esimerkiksi näkymien tunnistamiseen *fms:rootcat_id*-predikaatilla. Kun näkymät pystytään näin identifioimaan yksikäsitteisesti, voidaan esimerkiksi niiden taustavärit määrittellä helposti *Cascading Style Sheets*-kielellä (CSS) [CSS99]. Kaikkiin kategoriaan projektiovaiheessa liitettyihin ominaisuuksiin on mahdollista viitata arvolla *ogt:ANY_PROPERTY*. Kohteiden ominaisuuksia voi kysellä vastaavaan tapaan optiolla *ogt:incBookmarkProperty*. MuseoSuomen hakusivulla kohteiden tiedoista tarvitaan kuvatiedosto (*fms:www_kuvatiedostonimi*) ja kohteen alkuperäinen tunniste (*fms:www_numero*). Nimiö eli *rdfs:label* tulee vastaukseen aina automaattisesti kaikille resurssityyppeille arvoille.

Vastauksen toivottu kieli määritellään kyselyssä predikaatilla *ogt:acceptLanguage*. Kielimääritys koskee kaikkia vastaukseen tulevia literaaleja, joille kieli on määritelty. Jos kyselyssä ei määritellä kieltä, ei vastauksessakaan huomioida kielimäärityksiä. Tällöin, jos projektio sisältää useita kieliä, tulee kyselyssä aina määrittää kieli, jotta vältetään seka-kieliset vastaukset.

9.2.5 Rajoitteet

Ontogator mahdollistaa vastaukseen otettavien kategorioiden ja kohteiden määrällisen rajoittamisen. Rajoitteet ovat välttämättömiä, sillä rajoittamattoman vastauksen koko saattaa muuten kasvaa useisiin megatavuihin —jopa satoihin. Ontogatorissa on kahden tyyppisiä rajoitteita, joita käytetään kategorioiden ja kohteiden maksimimäärän rajoittamiseen (maksimirajoitteet) sekä ohittamiseen (ylihyppyrajoitteet). Rajoitteet kohdistuvat aina jonkin tietyn kategorian tai fasetin suorien alikategorioiden tai kohteiden määrään. Rajoitteet ovat luokan *ogt:Restriction* ilmentymiä. Niitä käytetään MuseoSuomessa kohteiden ja kategorioiden sivutukseen.

Maksimirajoitteet määritellään rakenteisesti siten, että varsinainen raja-arvo annetaan predikaatilla *ogt:limit* ja kohdistusmääre predikaatilla *ogt:onCategory*. Jos kohdistusmäärettä ei ole annettu, kohdistuu rajoite selektoria vastaavaan fasettiin. MuseoSuomen verkkäyttöliittymän näkymissä näytetään oletusarvoisesti korkeintaan kaksikymmentä kategoriata (kuva 7). Kaikki kategoriat saa näkyville ”näytä kaikki” -linkistä. Hakusivulla (kuva 9) kohteiden maksimimäärää rajoitetaan esikyselyn perusteella sen mukaan kuinka monta ryhmää tuloksessa on. Mitä vähemmän ryhmiä, sitä enemmän kohteita näytetään kustakin ryhmästä. Maksimirajoitteet koskevat annettua kategoriata tai fasettia ja periytyvät alikategorioille, jos näille ei ole määritelty erillisiä maksimirajoitteita.

Sekä kohteiden että kategorioiden sivutus ja/tai alisivutus on mahdollista toteuttaa käyttämällä maksimirajoitteiden ohella ylihyppyrajoitteita. MuseoSuomen hakusivulla sivutetaan sekä ryhmiä että ryhmiin liittyviä kohteita (kuva 9). *Ylihyppyrajoite* kohdistuu aina annettuun kategoriata tai fasettiin ja hyppää sen alikategorioiden (*ogt:skipSubCategories*) tai kohteiden (*ogt:skipBookmarks*) yli. Kuten maksimirajoitteetkin, määritellään ylihyppyrajoitteet rakenteisesti siten, että ylihyppättävien kategorioiden tai kohteiden määrä annetaan predikaatilla *ogt:limit* ja kohdistusmääre predikaatilla *ogt:onCategory*. Jos kohdistusmäärettä ei ole annettu, kohdistuu rajoite fasettiin. Ylihyppyrajoitteet koskevat aina annettua kategoriata tai fasettia eivätkä periydy tämän alikategorioille.

Hakuehtoon sopivien kohteiden (*ogt:incBookmarkHits*) ja kategorioiden (*ogt: incSubCa-*

categoryHits) määrien ohella hakutuloksen sivuttamista helpottaa optio *ogt:incPosition*. Se liittää hakutuloksen kategorioihin ja kohteisiin tiedon (*ogt:position*) siitä, kuinka mones kyseinen kategoria tai kohde on suhteessa muihin saman vanhemman kategorioihin tai kohteisiin. *ogt:position* riippuu hakuehdon rajoitteista ja huomioi myös erilaiset suodatus-tavat: *ogt:position*, kuten maksimi- ja ylihyppyrajoitteetkin, viittaa hakulogiikan kannalta suodattamattomiin kategorioihin, ei siis esimerkiksi harmaannuttamalla suodatettuihin.

9.3 Vastausrajapinta

Tässä aliluvussa kuvataan Ontogatorin vastausrajapinta yleisellä tasolla. Ontogator vastaa kyselyihin määrämuotoisessa RDF/XML-formaatissa. Määrämuotoisuudella taataan se, että vastauksia on mahdollisimman helppoa käsitellä XML-muodossa käyttäen esimerkiksi *XSL Transformation*-muunnosta [XSL99]. Peruseriaate on, että kaikki hierarkiat kirjoitetaan auki XML-muotoon.

Kuva 17 havainnollistaa kyselyn ja vastauksen suhdetta. Siinä on osa kuvassa 15 esitetyn näkymäehdon *bookmarksByCategory* vastauksesta. Alkuperäisen kyselyn (liite C) vastaus on kokonaisuudessaan liitteessä D. Kuva 9 perustuu tähän vastaukseen.

Vastaus käsittää ensisijaisesti *ogt:Facet*-luokan ilmentymiä eli fasetteja, yksi kutakin näkymäehto kohden. Fasetti viittaa sen tuottaneeseen näkymäehtoon *ogt:facetSelector*-predikaatilla. *Ogt:Facet* on *ogt:Category*-luokan erikoistapaus. Sen anonyymit instanssit luodaan aina täysin dynaamisesti kyselyn perusteella.

Kohteet liittyvät optioista ja mahdollisesta ryhmittelyehdosta riippuen joko suoraan fasettiin ja/tai sitten sen (ryhmittelyehdon määrittämiin) alikategorioihin. Kategorioihin liittyvät kohteet annetaan lista-arvoisesti predikaatilla *ogt:topicOf*. Kohteeseen liittyvät kategoriat puolestaan annetaan *bm:Bookmark*-luokan predikaatilla *bm:hasTopic*.

Ontogator pyrkii liittämään kaikkiin vastauksen sisältämiin resursseihin nimiön (*rdfs:label*). Jopa *ogt:facetSelector* -predikaatin arvoon liitetään nimiö, jos kyselyssä sellainen määritellään. Tämä toimintatapa nousee hakutavan käyttöliittymälähtöisyydestä: resurskien URI-tunnisteet eivät ole tarkoitettu käyttäjän luettaviksi. Nimiöt ovat käyttäjää varten.

9.4 Implementaatio

Ontogatorin julkinen API on määritelty Javan rajapintaluokkia käyttäen, joten syötteiden muodostus ja esitystapa on viimekädessä mahdollista optimoida tapauskohtaisesti.

```

<ogt:Facet> ▷ Anonyymi fasetti.
<ogt:facetSelector rdf:resource="&fmsweb;bookmarksByCategory"/> ▷ Fasetin tuottanut näkymäehto (ks. kuva 15).
<ogt:bookmarkHits>57</ogt:bookmarkHits> ▷ Annetuilla hakuehdoilla löytyy 57 erillistä kohdetta (incBookmarkHits = true).
<ogt:subCategories ▷ Kohteet ryhmiteltyinä kategorioittain (incCategories = true, incBookmarks = true)
  rdf:parseType="Collection">
  <ogt:Category ogt:catid="%00%0C%0F" ▷ Haettu kategoria, säilytysasiat (category = %00%0C%0F)
    rdfs:label="säilytysasiat">
    <ogt:subCategoryOf ogt:catid="%00%0C" rdfs:label="säilyttimet"/>
    <ogt:hasRoot>
      <ogt:Category ogt:catid="%00" rdfs:label="Esinetyyppi">
        <fms:rootcatid>0</fms:rootcatid> ▷ incCategoryProperty = fms:rootcatid
        <ogt:subCategories rdf:parseType="Collection">
          <ogt:Category ogt:catid="%00%0C" rdfs:label="säilyttimet"/>
        </ogt:subCategories>
      </ogt:Category>
    </ogt:hasRoot>
  </ogt:Category>
  <ogt:subCategories ▷ Kategorian säilytysasiat suorat alikategoriat (incSubCategories = DIRECT)
    rdf:parseType="Collection">
    <ogt:Category ogt:catid="%00%0C%0F%01" rdfs:label="kanisterit">
      <ogt:bookmarkHits>7</ogt:bookmarkHits>
      <ogt:subCategoryHits>0</ogt:subCategoryHits> ▷ Kategorialla kanisterit ei ole alikategorioita (incSubCategoryHits = true)
      <ogt:topicOf rdf:parseType="Collection"> ▷ Kategorian kanisterit seitsemästä kohteesta neljä (maxBookmarks ← [limit = 4])
        <bm:Bookmark rdf:about="&fms;ECM_2194_66" rdfs:label="kanisteri:taskumatti">
          <fms:www_numero>ECM:2194:66</fms:www_numero>
          <fms:www_kuvatiedostonimi>
            /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000581.png
          </fms:www_kuvatiedostonimi>
        </bm:Bookmark>
        <bm:Bookmark rdf:about="&fms;ECM_2199_168" rdfs:label="kanisteri:taskumatti">
          <fms:www_numero>ECM:2199:168</fms:www_numero> ▷ incBookmarkProperty = fms:www_numero
          <fms:www_kuvatiedostonimi> ▷ incBookmarkProperty = fms:www_kuvatiedostonimi
            /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000580.png
          </fms:www_kuvatiedostonimi>
        </bm:Bookmark>
        <bm:Bookmark rdf:about="&fms;ECM_2199_169" rdfs:label="kanisteri:taskumatti">
          ...
        </bm:Bookmark>
        <bm:Bookmark rdf:about="&fms;ECM_2423_253" rdfs:label="kanisteri:taskumatti">
          ...
        </bm:Bookmark>
      </ogt:topicOf>
    </ogt:Category>
    ...
  </ogt:subCategories>
  <ogt:subCategoryHits>8</ogt:subCategoryHits> ▷ Kategorialla säilytysasiat on yhteensä kahdeksan suodattamatonta alikategoriaa.
  <ogt:topicOf rdf:parseType="Collection"> ▷ Kategorian säilytysasiat suorat kohteet.
    <bm:Bookmark rdf:about="&fms;ECM_2037_37" rdfs:label="lamppu:umpikupu">
      ...
    </bm:Bookmark>
    <bm:Bookmark rdf:about="&fms;ECM_2978_3" rdfs:label="lamppu:umpikupu">
      ...
    </bm:Bookmark>
  </ogt:topicOf>
  <ogt:bookmarkHits>57</ogt:bookmarkHits> ▷ Kategoriaan säilytysasiat liittyy annetuilla hakuehdoilla yhteensä 57 erillistä kohdetta,
  <ogt:directBookmarkHits>2</ogt:directBookmarkHits> ▷ ...joista kaksi suoraan (incDirectBookmarkHits = true).
</ogt:Category>
</ogt:subCategories>
<ogt:subCategoryHits>1</ogt:subCategoryHits>
<ogt:topicOf rdf:parseType="Collection"/>
<ogt:remainder>0</ogt:remainder> ▷ Kaikki hakuehdon tuottamat kohteet on ryhmitelty (incRemainder = true).
</ogt:Facet>

```

Kuva 17: Osa Ontogatorin vastauksesta tavanomaiseen kyselyyn (kuva 15, liite C). Kuvan kommentissa viitataan fasetin tuottaneeseen näkymäehtoon (*bookmarksByCategory*, kuva 15). Vastaus kyselyyn on kokonaisuudessaan liitteessä D.

Ontogatorin oletustoteutus käyttää HP Labs:in Jena2 Semantic Web -ohjelmistokehystä RDF-muotoisten syötteiden käsittelyyn [HP 03]. Syötteitä käsitellään Jenan tietämyskantaa vastaavan Model-rajapintaluokan kautta, joten se voi olla käytännössä määritelty millä tahansa RDF-kielen sarjallistustavalla tai vaihtoehtoisesti se voi sijaita esimerkiksi tietokannassa.

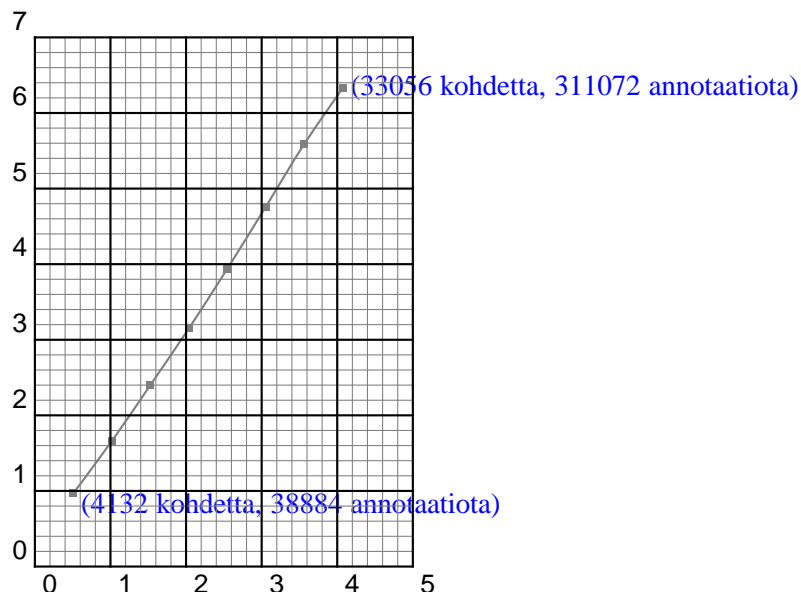
MuseoSuomessa käytetään projektiosyötteen käsittelyyn Model-rajapinnan muistinvarais-ta toteutusta. Projektiio määritellään ja luodaan Prologilla ja tallennetaan RDF/XML-muodossa tiedostoon, joka edelleen luetaan Ontogatoriin. Datan muuttuessa projektiio on luotava kokonaisuudessaan uudestaan.

Ontogator käyttää vastausten käsittelyyn *Simple API for XML* -rajapintaa¹⁹ (SAX). Se on tapahtumiin pohjautuva API XML:n käsittelyyn. Ontogatorin käyttäjä voi antaa kyselyn suorittavalle metodille oman `org.xml.sax.ContentHandler`-ilmentymän, joka käsittelee vastauksen. Se voi esimerkiksi serialisoida vastauksen XML-muotoisesti tai kuten MuseoSuomessa edelleen muokata vastausta esimerkiksi XHTML-muotoon [W3C02] käyttäen XSL-tyylitiedostoa [XSL99].

Ontogatorin kyselynsuoritusalgoritmi lähtee liikkeelle näkymäehdoista. Se laskee ensin kohteiden rajoitusehdon eli annettujen kohde-ehdojen tuottamien kohteiden leikkauksen. Tämän jälkeen, jos kategoriat halutaan vastaukseen, käydään kategoriaehdon löytämiä kategorioita läpi rekursiivisesti maksimissaan *incSubCategories*-option osoittamaan syvyyteen asti. Kunkin kategorian kohdalla algoritmi tarkistaa kuinka monta kategoriaan liittyvistä kohteista kuuluu kohteiden rajoitusehdon tuottamaan leikkaukseen. Jos 1) vähintään yksi kategoriaan liittyvistä kohteista kuuluu leikkaukseen (*ogt:filter = POST_QUERY*), 2) kategoriaan ylipäättään liittyy kohteita (*ogt:filter = PRE_QUERY*) tai 3) kategorioita ei lainkaan haluta suodattaa (*ogt:filter = NONE*), eikä kategoria leikkauksiin pois maksimi- tai ylihyppyrajoitteiden vuoksi, tulostetaan kategoria vastaukseen. Tämän jälkeen, jos kohteet halutaan vastaukseen, algoritmi tulostaa kohteet huomioiden maksimi- ja ylihyppyrajoitteet. Kategorioiden käsittelyn yhteydessä pidetään kirjaa kohteista, jotka ryhmittyvät kategorioittain. Lopuksi algoritmi tulostaa vastaukseen sellaiset leikkaukseen kuuluvat kohteet, jotka jäivät mahdollisen ryhmittelyn ulkopuolelle.

Ontogator käyttää kategorioiden identifioimiseen etuliiteskeeman [CPST03] mukaisia tunnisteita. *Etuliiteskeemassa* solmun tunnisteiden etuliitteenä on sen vanhemman tunniste. Esimerkiksi tunnisteella n identifioidun kategorian lapsien tunnisteet olisivat muotoa nm , jossa m on joko yksittäinen merkki tai määrämittainen merkkijono. Järjestämällä kategoriat sopivaan tietorakenteeseen tunnisteiden mukaisesti, voidaan annetun kategorian kaik-

¹⁹<http://www.saxproject.org>



Kuva 18: Kyselyjen suoritus aika (Y-akselilla sadasosasekunteina) suhteessa kategorioiden keskimääräisesti liittyvien kohteiden määrään ($\text{annotaatioita/kategorioiden}/10$). Kategorioiden kaikissa kuvan testeissä oli 7637.

ki alikategoriat hakea tehokkaasti. Ontogator käyttää sisäisesti vastaavaa indeksointitapaa myös kohteille. Kohdeindeksiä käytettäessä on kuitenkin huomioitava se, että sama kohde saattaa esiintyä alipuun useissa eri haaroissa. Hakutuloksesta nämä duplikaatit täytyy poistaa. Ontogator käyttää kategorioiden tunnistamiseen Javan char-tietotyyppiä, mikä asettaa tietyn kategorian suorien alikategorioiden ja kohteiden maksimimääräksi 2^{16} eli 65536.

Alustavien skaalautuvuustestien perusteella Ontogatorin vasteaika kasvaa lineaarisessa suhteessa kategorioiden liittyvien kohteiden määrään (kuva 18). Kategorioiden tai kategoriakohte-relaatioiden eli annotaatioiden määrät eivät testien perusteella sinänsä vaikuta kyselyjen suoritus aikaan, sillä kategorioiden määrän kasvattaminen kasvatti myös annotaatioiden määrää, eikä tällä ollut vaikutusta suoritus aikaan. Kohteiden määrän ei sinänsä periaatteessa pitäisi vaikuttaa muihin kuin sanahakuihin. Kategoriaperustaisissa haussa puolestaan ratkaisevaa on annotaatioiden määrä. Jos kategoriaan liittyvien kohteiden määrä ajatellaan tämän painona, pitäisi kohteita pystyä lisäämään järjestelmään ilman, että tämä vaikuttaa olennaisesti suoritus aikaan, jos samalla lisätään kategorioiden ja tarvittaessa myös juurikategorioiden määrää siten, ettei yksittäisten kategorioiden painot kasva liikaa.

Testiaineistoa generoitiin käsittelemällä MuseoSuomen RDF/XML-muotoista syötettä siten, että 1) kategori hierarkiat monistettiin annotaatioineen ja 2) kohteet ja annotaatiot monistettiin. Testiajoja suoritettiin erilaisilla $n * m$ kombinaatioilla, jossa n on alkupe-

räisen aineiston kategorioiden ja m kohteiden määrän monikerta. Yllättävää testeissä oli ennen kaikkea se, ettei kategorioiden määrä sinänsä vaikuttanut suoritusajkaan, vaikka samalla myös kategoria–kohde-suhteiden määrä kasvoi.

Testien lähtöaineistona käytettiin MuseoSuomi-järjestelmän sen hetkistä (20.4.2004) aineistoa. Kategoriahierarkioiden maksimisyvyys oli kymmenen ja keskimääräinen syvyys 4.52. Testikyselyinä käytettiin viittätoista MuseoSuomessa lokien perusteella usein toistuvaa toisistaan poikkeavaa kyselyä. Analyysi perustui näiden viidentoista kyselyn keskimääräiseen suoritusajkaan. Satunnaisesti suoritettavan useita sadasosasekunteja vievän roskien keruun vaikutukset suoritusajkaan minimoitiin kutsumalla roskien keruuta (`System.gc()`) eksplisiittisesti aina kyselyiden välillä.

Testeissä käytettiin Intel Pentium 4 2.80GHz -suorittimella varustettua Linux-konetta, jossa oli muistia 4G. Muistista tosin saatiin käyttöön Javan virtuaalikoneen rajoituksista johtuen vain 1.8G. Virtuaalikoneena käytettiin Java HotSpot(TM) Client VM -virtuaalikonetta (build 1.4.2_01-b06, mixed mode).

10 Ratkaisun arviointia

Tässä luvussa arvioidaan esitettyjä ratkaisuja. Ensin esitetään kokemuksia RDF-kielestä (luku 10.1), sitten arvioidaan projektioskeeman toimivuutta sekä projektioiden toteutusta MuseoSuomi-järjestelmässä (luku 10.2). Tämän jälkeen arvioidaan Ontogatorin kyselyrajapintaa ja lopuksi varsinaista implementaatiota (luku 10.3).

10.1 RDF sovelluskehityksessä

RDF on perusteiltaan intuitiivinen ja helppokäyttöinen kieli. Jena2 [HP 03] on osoittautunut varsin käyttökelpoiseksi ja monipuoliseksi RDF-rajapinnaksi. RDF-kielen käyttöönottoa helpottaa myös ihmisen kannalta vaikealukuinen RDF/XML-sarjallistusmuoto, joka mahdollistaa XML-kielelle luotujen tekniikkojen (kuten XSLT) ja ohjelmistojen käytön. Tällöin on kuitenkin huomioitava, että sarjallistuksen tulee olla määrämuotoista, sillä RDF/XML-kielellä voi saman asian ilmaista hyvin monella vaihtoehtoisella tavalla.

Suurin ongelma RDF-kielen käyttöönotossa on tällä hetkellä datan tuottamiseen liittyvät ongelmat. Käytettävissä olevat ilmaiset RDF-kieltä osaavat tietämuskantaeditorit ovat monin tavoin puutteellisia. Relaatiotietokannoista tiedon siirto semanttisesti rikkaampaan RDF-muotoon onnistuu harvoin ainakaan täysin automaattisesti. Avainkysymykseksi siir-

rossa nousevat avainsanat, joita ei ole käytetty systemaattisesti. Tämä aiheuttaa ongelmia paitsi homonyymien ja synonyymien myös kirjoitusvirheiden vuoksi. Hakumoottorin toteutuksessa päänvaivaa aiheuttavat erityisesti anonyymit solmut, sillä niihin ei ole mahdollista viitata ne määrittelevän tietämuskannan ulkopuolelta.

10.2 Projektioista

Esitetyssä projektioskeemassa muuten samanlaiset kategoriahierarkiat tulkitaan erillisiksi, jos niihin liittyvät eri kohteet. Tämän vuoksi esimerkiksi paikka- ja toimijahierarkiat ovat MuseoSuomen kategoriaprojektiossa kahteen otteeseen, käyttö- ja valmistusnäkökulmista. Osin tästä johtuen nykyisen kaltaisen kategoriaprojektion koko voi kasvaa jopa alkuperäistä tietämuskantaa suuremmaksi. Jotta järjestelmä skaalautuisi isoihin aineistoihin, tulisi projektioskeemaa muuttaa. Projektioiden tulisi olla mahdollisimman kevyitä ja mahdollisesti täysin kyselyn yhteydessä dynaamisesti luotuja eikä esimerkiksi nykyisen kaltainen nimiöiden liittäminen projektiioon tulisi kysymykseen.

MuseoSuomessa projektiosäännöt esitetään ja suoritetaan Prolog-logiikkaohjelmointikielellä. Siinä missä tämä lähestymistapa mahdollistaa mielivaltaisen monimutkaiset säännöt, olisi projektiosääntöjen esittämisessä ensisijaisesti RDF-kielellä kuitenkin lukuisia etuja tähän verrattuna:

1. Säännöt voisi luoda ja niitä voisi muokata samalla tietämuskannan hallintaohjelmalla, jolla itse dataa käsitellään.
2. Ontologian rakentamisen yhteydessä voisi tällöin helposti määritellä myös ontologian käyttötavan näkymäpohjaisessa haussa. Toisaalta vastaavaa hierarkioiden visualisointia voisi hyödyntää myös itse tietämuskannan hallintaohjelmassa. Tällöin tietämuskantaan tehtyjen muutosten vaikutus kategorianäkymiin olisi ilmeinen ja dataa voisi muokata suoraan hierarkian muodossa.
3. Mahdolliset luokkien ja ominaisuuksien nimien tai nimiavaruuksien eli URI-tunnisteiden muutokset heijastuisivat automaattisesti myös hierarkiasääntöihin. Näin vältettäisiin hierarkian rakenteen kuvaavien hierarkiasääntöjen ja ontologian URI-tunnisteiden synkronointiongelmat. Esimerkiksi Promootorissa URI-tunnisteiden synkronointiongelmat olivat jatkuvana riesana, sillä luokkien ja ominaisuuksien URI-tunnisteet muuttuivat usein aineiston päivityksen yhteydessä. Vastaavat muutokset täytyi tehdä käsin myös konfiguraatitiedostoihin sekä projektiio- ja suositussääntöihin. Tietämuskannan ylläpitämiseen käytetty tietämuskannan hallintaohjel-

ma puolestaan huolehti tietämuskannan sisäisten viittausten eheydestä automaattisesti.

4. Perushierarkiasääntöjen esittämiseen ei vaadittaisi esimerkiksi Prologin tai muun ohjelmointi- tai sääntökielen osaamista. Vain erikoistapausten käsittely vaatisi ohjelmoijan panosta.
5. Hierarkiasääntöjen käsittely ei olisi sidottu mihinkään tiettyyn ohjelmointi- tai sääntökieleen. Toistaiseksi Semantic Computing -tutkimusryhmässä toteutetuissa näköpohjaisissa hakusovelluksissa projektiosääntöjen esittämiseen on käytetty ohjelmointikieliä kuten Java ja Prolog.

Koska yleisessä tapauksessa projektiosäännöt voivat olla mielivaltaisen monimutkaisia, on geneerisen järjestelmän viimekädessä mahdollistettava sääntöjen määrittäminen myös jollain ohjelmointi- tai sääntökielellä, kuten Prologilla, CWM:lla²⁰, Jenan sääntökielellä [HP 03] tai Javalla. Siinä missä täysin ilmaisuvoimaisen RDF-pohjaisen projektiosääntökielen toteuttaminen ja käyttäminen olisi erittäin haastavaa ja monimutkaista, eikä siksi edes tarkoituksenmukaista käyttäjän kannalta, ovat yleiset perustapaukset yksinkertaisia. Toisaalta monimutkaisetkin tapaukset olisi mahdollista toteuttaa esimerkiksi siten, että erityyppiset projektiosäännöt identifioitaisiin URI-tunnisteella ja itse säännöissä annettaisiin kaikki kyseisen säännön suorituksessa tarvittavat vapaat parametrit. Käytettävä projektioalgoritmi valittaisiin säännön tyyppin URI-tunnisteen perusteella. Samalle säännölle voisi tällöin olla useita erikielisiä toteutuksia ja varsinaiset monimutkaiset säännöt olisi mahdollista kuvata parhaaksi katsotulla kielellä RDF-kuvauksien pysyessä yksinkertaisina.

Hierarkiaprojektoiden perustapaukset on mahdollista esittää RDF-muotoisesti suoraan luvussa 8.1 esitetyllä hierarkiasääntöskeemalla (esimerkki 9). Skeeman mukaisia hierarkiasääntöjä käsittelee algoritmi PROJECT-3. Kohdeprojektiosääntöjen RDF-notaatio voisi esimerkiksi olla suora käänös RDF-polusta, eli lista predikaattimäärittäyksiä, joiden suunnan voi määritellä. Tällaisenaan kukin polkumäärittäyksen elementti vastaisi läheisesti hierarkiasääntöjen kaarimäärittäystä (:ArcDefinition, esimerkki 9).

Toisaalta esimerkiksi kaikki Promootorissa olevat kohdeprojektiosäännöt ovat niissä esiintyvien predikaattien suhteen assosiatiivisia. Esimerkiksi *promootio: mihin_instansseihin_liittyy* assosioi aina objektina olevat resurssit kuvaan ja samoin *promootio: roolissa_olevat_henkilöt* assosioi henkilö-resurssin aina vastaavaan rooliin ja edelleen tätä kautta kuvaan. Tämän kaltaiset kohdeprojektiosäännöt olisikin helpompi esittää

²⁰<http://www.w3.org/2000/10/swap/doc/cwm.html>

joukkona transitiivisesti ristiin assosiativisia suhteita: henkilö-resurssit assosioituvat aina rooleihin, tiedekunnat promootiotiedekuntiin ja nämä puolestaan promootiotapahtumiin. Näin vältettäisiin monimutkaisten ja ainakin osittain toisteisten polkujen tarve.

Ongelmallista esitetyissä kohdeprojektiosäännöissä on primäärisen arvon tunnistaminen. Ei esimerkiksi ole käyttäjän kannalta mielekäästä esittää kuvassa olevan sekä rehtori Kari Raivio (primäärinen arvo) että tämän roolin kautta kuvaan assosioitunut henkilö Kari Raivio. Tämä on ihmiselle itsestään selvää.

10.3 Ontogator-hakumoottorista

Ontogatorin kyselyrajapinnan kategoria-, kohde- ja näkymäehtojen suhdetta tulisi selkeyttää. Rakenne olisi huomattavasti selkeämpi, jos ne olisivat erillisiä luokkia. Esitetyssä rajapinnassa esimerkiksi kategoriaehdon tulkinta riippuu sen roolista kyselyssä, eikä tämä välttämättä ole ilmeinen. Kategoriaehto voidaan käyttää niin kategorioiden hakuun, kohteiden ryhmittelyyn kuin kohteiden hakuunkin. Selkeästi näkymä- ja kohde-ehdoista erillistä kategoriaehto voisi toisaalta käyttää sellaisenaan myös maksimi- ja ylihyppyrajoitteidenkin kohdistusmääreenä siinä missä nykytoteutuksessa kohdistusmääre viittaa kategoriatunnisteseen ja on täysin erillinen muista kategoriaehdoista. Lisäksi nykyrajapinnalla kategoria on tunnettava, jotta siihen on mahdollista kohdistaa rajoitteita.

Suurin puute Ontogatorin kyselyrajapinnassa on kuitenkin yleisten ominaisuusrajoitteiden puute. Vasta niiden myötä generisen näkymäpohjaisen hakumoottorin ilmaisuvoimaa pystyisi todella laajentamaan tavalla, jonka RDF mahdollistaa yleisellä tasolla laajassa mittakaavassa. Periaatteessa ominaisuusrajoitteilla voisi ilmaista myös kategoria ja kohde-ehdot. Esimerkiksi kategoriaehdon *ogt:category*-osan voisi toteuttaa kategorialuokan *ogt:catid*-ominaisuuteen kohdistuvana ominaisuusrajoitteena. Koska tällaisia ominaisuusrajoitteita ei kuitenkaan toistaiseksi ole implementoitu yhdessäkään järjestelmässä, vaativat ne lisätutkimusta niin käytettävyyden, hakurajapinnan kuin toteutuksen algoritmienkin osalta.

Ontogatorin API-rajapinnassa erityisen käyttökelpoiseksi ratkaisuksi Web-sovelluksen kannalta osoittautui SAX-rajapinnan käyttö vastauksien tulostamiseen. Se mahdollistaa esimerkiksi Ontogatorin tehokkaan käytön osana XML-transformaatioista koostuvaa putkea XSLT-muunnosten ohella.

Vaikka vastauksen kieli onkin mahdollista määritellä kyselyssä ja kaikkiin resursseihin liitetään mahdollisuuksien mukaan nimiöt, ei monikielistä dynaamista käyttöliittymää ole mahdollista toteuttaa yksin Ontogator-pohjaisesti. Jotta esimerkiksi kohteiden tiedot olisi

mahdollista esittää dynaamisesti käyttöliittymässä, ei riitä, että resurssien (predikaattien arvojen) nimiöt tiedetään; myös itse predikaattien ontologiassa määritellyt nimiöt on tunnettava.

Ontogatorin vastaus sisältää tällä hetkellä vain käyttäjän pyytämät fasetit ja Ontogatoriin ladataan vain projektiot. Jos Ontogatoriin ladattaisiin projektioden lisäksi myös projektioden taustalla vaikuttavat ontologiat, olisi suhteellisen yksinkertaista toteuttaa sellainen lisäoptio, joka liittäisi esimerkiksi vastausdokumentin loppuun siinä käytettyjen ominaisuuksien nimiöt. Tällä hetkellä Ontogator-pohjainen käyttöliittymä on rakennettava perinteisten informaatiosovellusten tapaan siten, että kenttien nimet ovat kovakoodattuina käyttöliittymässä. Toinen vaihtoehto olisi käyttää erillistä RDF-palvelinta ontologiakyselyille.

Vaikka Ontogatorin implementaatiossa olisi paljonkin optimoimisen varaa sellaisenaankin, on skaalautuvuuden kannalta suurin ongelma itse algoritmi, joka ei sellaisenaan sovellu isoihin aineistoihin. Laajan aineiston kannalta on kestävämpi vaihtoehto laskea ensin kohteiden rajoitusehto ja sitten verrata ryhmittelyehdon kategorioihin liittyviä kohteita tähän. Tässä joudutaan pahimmillaan tekemään paljon turhaa laskentaa varsinkin, mikäli vain harva tai ei yksikään kategorioihin liittyvistä kohteista kuulu leikkaukseen. Tietokantapohjaisessa lähestymistavassa tehokkainta olisi suorittaa kategorioiden ja niihin annetuilla ehdoilla liittyvien kohteiden haku yhdellä ja samalla kyselyllä. Toisaalta siinä missä nykyisellä lähestymistavalla erilaisten suodatusvaihtoehtojen toteuttaminen ja ryhmittelyyn kuulumattomien kohteiden hakeminen on kohtalaisen suoraviivaista, olisi näiden toteuttaminen tämänkaltaisella tietokantapohjaisella lähestymistavalla jokseenkin hankalaa.

Ontogatorin kategoriatunnisteskeema on valittu lähinnä implementaation helppouden nojalla. Tietokantapohjaisessa lähestymistavassa etuliiteskeeman mukaisten kategoriatunnisteiden käyttäminen ei kuitenkaan ole kyselyiden kannalta paras mahdollinen vaihtoehto. Kyselyiden kannalta käytännöllisempi vaihtoehto voisi olla esimerkiksi kategorioiden tunnistaminen jonkin intervalliskeeman mukaisesti [CPST03]. Intervalliskeemoissa kategoria tunnistetaan lukuparilla (*alku, loppu*), jotka määrittävät kategorian paikan hierarkiassa kahdesta suunnasta, esimerkiksi esi- ja jälkijärjestyksessä. Intervalliskeemoissa hierarkian päivittäminen vaatii usean solmun tunnisteen muuttamista. Web-käyttöliittymää ajatellen ei toisaalta mikään dynaamisesti määräytyvä tunnisteskeema ole hyvä, sillä tunnisteen muuttuminen rikkoo myös käyttöliittymän linkit. Ainoa varma tapa tunnistaa kategoriatiot olisi niihin viittaaminen niiden täydellisten (projisoitujen) polkujen avulla. Käytännössä ilman tehokasta URL-yhteensopivaa URI-tunnisteiden tiivistystapaa tällaiset tun-

nisteet olisivat kuitenkin pituutensa vuoksi todennäköisesti käyttökelvottomia.

11 Yhteenveto

Tässä tutkielmassa on osoitettu kuinka näkymäpohjaista hakua voidaan soveltaa ja edelleen laajentaa RDF-kieleen pohjautuen. Näkymäpohjainen haku on laajoissa käytettävyytutkimuksissa todettu käytettävyyden kannalta hyväksi. Siinä yhdistyy helppokäyttöisyys hyvään löytyvyyteen ja tarkkuuteen. Korkean löytyvyyden ja tarkkuuden mahdollistavat käsittehierarkiat, joita käytetään myös haun muodostamisessa ja tulosten esittämisessä.

Näkymäpohjaisen haun vaatimaa työlästä hierarkioissa navigoimista on mahdollista vähentää sanahaualla. Sanahaku kohdistuu sekä kategorioihin että kohteisiin. Sanahaun löytämien kategorioiden perusteella hakua on mahdollista tarkentaa sanan tarkoitetun merkityksen suhteen.

Ominaisuusrajoitteet mahdollistavat näkymäpohjaisen haun yhdistämisen saumattomasti perinteiseen lomakepohjaiseen hakuun. Tämä on tarpeellista, koska kaikilla käsitteillä ei ole näkymäpohjaisen haun vaatimaa luonnollista hierarkkista esitysmuotoa tai se on liian rajoittava. Eräs tällainen käsite on aika: vuosisatoihin, -kymmeneen ja vuosiin perustuva hierarkia ei mahdollista esimerkiksi avoimeen aikavälin tai aikakausien vaihteisiin kohdistuvia hakuja. Myös sanahaku olisi mahdollista toteuttaa ominaisuusrajoitteena. Tällöin kyselyssä olisi mahdollista määrittää ominaisuudet, joihin sanahaku todella kohdistuu. Tällä hetkellä sanahaussa käytettävät ominaisuudet määritellään konfiguraatiossa.

Näkymäpohjainen haku perustuu kielitieteellisiin käsitesuhteisiin kuten hyponymiaan ja meronymiaan. Semantic Webin tietämyksen esittämistapa puolestaan pohjautuu semanttisiin verkkoihin, kehyspohjaisiin järjestelmiin ja formaaliin logiikkaan. Siinä missä näkymäpohjaisen haun kategoriahierarkiat ovat ortogonaalisia puita, muodostavat RDF-kuvaukset mielivaltaisia verkkoja. Näkymäpohjainen RDF-haku kohdistuu kategorioiden osalta alkuperäisen tietämyskannan projektioon. Projektiota käytetään peittämään alakohdittaiset ontologia- ja annotaatiomallinnusratkaisut. Näin varsinaisen hakumoottorin toteutus yksinkertaistuu ja yleistyy.

Ontogator on yleiskäyttöinen näkymäpohjainen RDF-hakumoottori. Sitä on käytetty muun muassa MuseoSuomi-järjestelmän hakukoneessa. Ontogatorin kaikki syötteet ja tulosteet ovat RDF-muotoisia. Se saa valmiiksi lasketut projektiot syötteenä, joten varsinainen projektiot on mahdollista suorittaa millä hyvänsä ohjelmointikielellä. Toisaalta tämä moni-

mutkaistaa tietojen päivittämistä ja käytännössä pienikin muutos vaatii koko projektion uudelleen laskemista ja lataamista Ontogatoriin.

Ontogatorin kyselyrajapinta määrittää erikseen kaikki mahdolliset rajoitetyypit. Tämä helpottaa tehokkaan implementaation tekemistä: kun tiedetään mihin ominaisuuksiin haku voi kohdistua, voidaan näille laskea tehokkaat hakemistot valmiiksi. Kohteisiin ja kategorioihin erikseen kohdistettavissa olevan sanahaun lisäksi Ontogatorin kyselyrajapinta määrittää yhden kategorioihin kohdistuvan ominaisuusrajoitteen: kategorioita on mahdollista hakea sen perusteella, mistä resurssista ne on projisoitu. Tätä ominaisuusrajoitetta hyödynnetään esimerkiksi MuseoSuomen mobiilikäyttöliittymässä: erillinen paikannuspalvelu palauttaa resursseja, joiden lähellä käyttäjä on, ja näitä käytetään edelleen hakuperusteina Ontogator-kyselyssä.

Ontogatorin jatkokehityksessä tulisi ominaisuusrajoitteiden ja tietokantapohjaisuuden ohella huomioida myös projektiosääntöjen ensisijainen esittäminen RDF-kielellä. Tämä helpottaisi tietojen määrittämistä ja päivittämistä sekä mahdollistaisi projisointiin käytetyn kielen vaihtamisen ilman tapauskohtaisten sääntöjen uudelleen kirjoittamista.

Lähteet

- BG04 Brickley, D. ja Guha, R., RDF vocabulary description language 1.0: RDF Schema, 2004. URL <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>.
- BLSC03 Berners-Lee, T., Sandro ja Connolly, D., Semantic Web tutorial using N3, 2003. URL <http://www.w3.org/2000/10/swap/doc/>.
- BPT02 Buntine, W., Perttu, S. ja Tirri, H., Building and maintaining web taxonomies. *Towards the Semantic Web and Web Services — Proceedings of the XML Finland 2002 Conference*, Hyvönen, E. ja Klementtinen, M., toimittajat. XML Finland, HIIT Publications, October 2002, sivut 54–65.
- BvHH⁺04 Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F. ja Stein, L. A., OWL Web Ontology Language reference, 2004. URL <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>.
- BYRN99 Baeza-Yates, R. A. ja Ribeiro-Neto, B. A., *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999. URL citeseer.nj.nec.com/433337.html.
- CP99 Cranefield, S. ja Purvis, M., UML as an ontology modelling language. *Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*. CEUR Publications, 1999, sivut 46–53, URL <http://citeseer.nj.nec.com/cranefield99uml.html>.
- CPST03 Christophides, V., Plexousakis, D., Scholl, M. ja Tourtounis, S., On labeling schemes for the semantic web. *Proceedings of the twelfth international conference on World Wide Web*. ACM Press, 2003, sivut 544–555.
- CSS99 Cascading Style Sheets, level 1, 1999. URL <http://www.w3.org/TR/CSS1>.
- DAM DARPA Agent Markup Language (DAML). URL <http://www.daml.org/about.html>.

- DAM01 Reference description of the DAML+OIL (march 2001) ontology markup language, 2001. URL <http://www.daml.org/2001/03/daml+oil-index>.
- Fel98 Fellbaum, C., toimittaja, *WordNet — An Electronic Lexical Database*. The MIT Press, London, England, 1998.
- Fen03 Fensel, D., *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, New York, Inc., 2003.
- FLGD87 Furnas, G. W., Landauer, T. K., Gomez, L. M. ja Dumais, S. T., The vocabulary problem in human-system communication. *Commun. ACM*, 30,11(1987), sivut 964–971.
- GMM03 Guha, R., McCool, R. ja Miller, E., Semantic search. *Proceedings of the twelfth international conference on World Wide Web*. ACM Press, 2003, sivut 700–709.
- Gru93 Gruber, T. R., A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5,2(1993), sivut 199–220. URL <http://dx.doi.org/10.1006/knac.1993.1008>.
- HB96 Henninger, S. ja Belkin, N. J., Interface issues and interaction strategies for information retrieval systems. *Conference Companion on Human Factors in Computing Systems*. ACM Press, 1996, sivut 352–353, URL <http://doi.acm.org/10.1145/257089.257364>.
- HEE⁺02 Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K. ja Yee, K.-P., Finding the flow in web site search. *Communications of the ACM*, 45,9(2002), sivut 42–49. URL <http://doi.acm.org/10.1145/567498.567525>.
- Hef01 Heflin, J. D., *Towards The Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment*. Väitöskirja, University of Maryland, College Park, 2001.
- HFB⁺00 Horrocks, Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., F. van Harmelen 2, a. M. K., Staab, S., Studer, R. ja Motta, E., The Ontology Inference Layer OIL. Tekninen raportti, On-To-Knowledge, <http://www.ontoknowledge.org>, 2000. URL <http://www.cs.vu.nl/~dieter/oil/Tr/oil.pdf>.

- HJK⁺04 Hyvönen, E., Junnila, M., Kettula, S., Mäkelä, E., Saarela, S., Salminen, M., Syreeni, A., Valo, A. ja Viljanen, K., Publishing museum collections on the Semantic Web —the MuseumFinland portal. *Proceedings of the thirteenth international conference on World Wide Web*. ACM Press, 2004.
- HP 03 HP Labs Semantic Web Research, Jena, 2003. URL <http://jena.sourceforge.net/>.
- HSV04 Hyvönen, E., Saarela, S. ja Viljanen, K., Application of ontology techniques to view-based semantic search and browsing. *The Semantic Web: Research and Applications. Proceedings of the First European Semantic Web Symposium (ESWS 2004)*, Bussler, C., Davies, J., Fensel, D. ja Studer, R., toimittajat. Springer-Verlag, 2004, URL <http://www.cs.helsinki.fi/u/eahyvone/publications/promoottori%.pdf>.
- KH02 Kivelä, A. ja Hyvönen, E. *Semantic Web Kick-Off in Finland — Visions, Technologies, Research, and Applications*, luku Ontological Theories for the Semantic Web, sivut 111–136. HIIT Publications, 2002.
- Lei89 Leiwo, M. *Lapsen kielen kehitys*, luku 1, sivu 9. Yliopistopaino, Helsinki, 1989.
- LS99 Lassila, O. ja Swick, R., Resource Description Framework (RDF) model and syntax specification, 1999. URL <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>.
- MS00 Markkula, M. ja Sormunen, E., End-user searching challenges indexing practices in the digital newspaper photo archive. *Information Retrieval*, 1,4(2000), sivut 259–285.
- NFM00 Noy, N. F., Fergerson, R. W. ja Musen, M. A., The knowledge model of Protégé-2000: Combining interoperability and flexibility. *2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*, 2000, URL http://www-smi.stanford.edu/pubs/SMI_Reports/SMI-2000-0830.pd%fc.
- Obj02 Object Management Group (OMG), MetaObject Facility (MOF) specification, version 1.4, 2002. URL <http://www.omg.org>.
- OR23 Ogden, C. K. ja Richards, I. A., *The Meaning of Meaning*. Hartcourt, Brace, and World, New York, 8. painos, 1923.

- OWL04 OWL Web Ontology Language use cases and requirements, 2004. URL <http://www.w3.org/TR/2004/REC-webont-req-20040210/>.
- Pol97 Pollitt, S. A., The key role of classification and indexing in view-based searching. *Proceedings of the 63rd IFLA General Conference*, 1997, URL <http://www.ifla.org/IV/ifla63/63polst.pdf>.
- RDF01 W3C RDF validation service, 2001. URL <http://www.w3.org/RDF/Validator/>.
- SDWW01 Schreiber, A., Dubbeldam, B., Wielemaker, J. ja Wielinga, B., Ontology-based photo annotation. *Intelligent Systems, IEEE*, osa 16. IEEE, 2001, sivut 66–74, URL <http://ieeexplore.ieee.org/iel5/5254/20350/00940028.pdf?isNum%ber=20350?=JNL&arnumber=00940028>.
- Sow00 Sowa, J. F., *Knowledge Representation — Logical, Philosophical, and Computational Foundations*. Brooks/Cole, California, USA, 2000.
- SW01 W3C Semantic Web activity, 2001. URL <http://www.w3.org/2001/sw/>.
- SWM04 Smith, M. K., Welty, C. ja McGuinness, D. L., OWL Web Ontology Language guide, 2004. URL <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>.
- TS888 *Sanastotyön käsikirja, Soveltavan terminologian periaatteet ja työmenetelmät*, osa SFS-käsikirja 50. Gummerus Kirjapaino OY, 1988.
- VES00 VESA - verkkosanasto, 2000. URL <http://vesa.lib.helsinki.fi/>.
- W3C02 W3C HTML Working Group, XHTML 1.0 The Extensible HyperText Markup Language (second edition), 2002. URL <http://www.w3.org/TR/xhtml1/>.
- WKLW98 Weibel, S., Kunze, J., Lagoze, C. ja Wolf, M., Dublin Core Metadata for resource discovery. Internet RFC 2413., 1998. URL <http://www.ietf.org/rfc/rfc2413.txt>. Ks. myös Dublin Core Metadata Initiative (DCMI) <http://dublincore.org/>.
- WSWS01 Wielinga, B. J., Schreiber, A. T., Wielemaker, J. ja Sandberg, J. A. C., From thesaurus to ontology. *Proceedings of the international conference on Knowledge capture*. ACM Press, 2001, sivut 194–201, URL <http://doi.acm.org/10.1145/500737.500767>.

- XML01 XML Schema part 2: Datatypes, May 2001. URL <http://www.w3.org/TR/xmlschema-2/>.
- XSL99 XSL Transformations (XSLT) version 1.0, 1999. URL <http://www.w3.org/TR/xslt.html>.
- YSLH03 Yee, K.-P., Swearingen, K., Li, K. ja Hearst, M., Faceted metadata for image search and browsing. *Proceedings of the conference on Human factors in computing systems*. ACM Press, 2003, sivut 401 – 408, URL <http://doi.acm.org/10.1145/642611.642681>.

A Bookmark-ontologia

W3-konsortiumin Annotea-projektin Bookmark-ontologia (<http://www.w3.org/2003/07/Annotea/BookmarkSchema-20030707>).

```
<?xml-stylesheet href="http://www.w3.org/2002/06/rdfs2html.xsl" type="text/xsl"?>
<rdf:RDF
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <rdf:Description rdf:about="">
    <rdfs:comment>$Id: bookmark.rdf,v 1.16 2003/07/18 18:15:18 swick Exp $</rdfs:comment>
    <dc:title>Bookmark Schema</dc:title>
    <dc:description>The Annotea Bookmark Schema, describing properties used
      to define instances of bookmarks, topics, and shortcuts.</dc:description>
  </rdf:Description>

  <rdf:Description rdf:about='http://www.w3.org/2002/01/bookmark#hasTopic'>
    <rdfs:comment>relates a bookmark to a topic.
      A bookmark must have at least one hasTopic property.
      The typical user operation of following a bookmark link
      will use the value of the b:recalls property. This property
      corresponds to XBEL:href property.</rdfs:comment>
    <rdf:type rdf:resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property' />
    <rdfs:label>hasTopic</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about='http://www.w3.org/2002/01/bookmark#leadsTo'>
    <rdfs:comment xml:lang='en'>connects a Shortcut to the bookmark or topic that is
    being included by reference in some other topic</rdfs:comment>
    <rdf:type rdf:resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property' />
  </rdf:Description>

  <rdf:Description rdf:about='http://www.w3.org/2002/01/bookmark#Topic'>
    <rdf:type rdf:resource='http://www.w3.org/2000/01/rdf-schema#Class' />
    <rdfs:label>Topic</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about='http://www.w3.org/2002/01/bookmark#Shortcut'>
    <rdfs:comment>Specifies a behavior; when the
      object of type 'Shortcut' is activated, the
      client follows the 'recalls' property and
      activates the object at the end of that
      'recalls' property. The target object may
      be another Bookmark or may be a Topic.</rdfs:comment>
    <rdf:type rdf:resource='http://www.w3.org/2000/01/rdf-schema#Class' />
    <rdfs:label>Shortcut</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about='http://www.w3.org/2002/01/bookmark#bookmarks'>
    <rdfs:comment>This corresponds to XBEL:href an
      object of type Bookmark is expected to have a
      'recalls' relationship to the document being
      bookmarked. The 'bookmarks' property is an older
      name for the 'recalls' relationship.</rdfs:comment>
    <rdf:type rdf:resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property' />
    <rdfs:label>bookmarks</rdfs:label>
  </rdf:Description>

  <rdf:Description rdf:about='http://www.w3.org/2002/01/bookmark#recalls'>
    <rdfs:comment>Relates a bookmark with the resource that
      has been bookmarked. This corresponds to XBEL:href;
      an object of type Bookmark is expected to have a
      'recalls' relationship to the document being
      bookmarked </rdfs:comment>
```

```
<rdf:type rdf:resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property' />
<rdfs:label>recalls</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about='http://www.w3.org/2002/01/bookmark#subTopicOf'>
  <rdfs:comment>Describes a relationship between Topics.
  When a topic T is a sub-topic of a topic U then all
  bookmarks that have topic T are also considered to have
  topic U. A topic may be a sub-topic of one or more
  topics; trivially, every topic is a sub-topic of itself.
  More formally; for all B, T, and U: b b:hasTopic T,
  T b:subTopicOf U implies B b:hasTopic U.</rdfs:comment>
  <rdf:type rdf:resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property' />
  <rdfs:label>subTopicOf</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about='http://www.w3.org/2002/01/bookmark#Bookmark'>
  <rdf:type rdf:resource='http://www.w3.org/2000/01/rdf-schema#Class' />
  <rdfs:label>Bookmark</rdfs:label>
  <rdfs:comment>The class to which all bookmarks belong.</rdfs:comment>
</rdf:Description>

</rdf:RDF>
```

B Ontogator-ontologia

Ontogatorin syötteet ja vastaukset kuvaava ontologia.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <!ENTITY bm 'http://www.w3.org/2002/01/bookmark#'>
  <!ENTITY ogt 'http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#'>
  <!ENTITY xsd 'http://www.w3.org/2000/10/XMLSchema#'>
  <!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
  <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
]>
<rdf:RDF xml:base="&ogt;"
  xmlns="&ogt;"
  xmlns:ogt="&ogt;"
  xmlns:bm="&bm;"
  xmlns:owl="&owl;"
  xmlns:rdf="&rdf;"
  xmlns:xsd="&xsd;"
  xmlns:rdfs="&rdfs;">

  <owl:Ontology rdf:about="">
    <rdfs:comment>$Id: ontogator.owl,v 1.1.2.1 2004/03/18 09:51:03 ssaarela Exp $</rdfs:comment>
    <rdfs:label>View-based search ontology for Ontogator.</rdfs:label>
  </owl:Ontology>

  <owl:Class rdf:ID="CategoryBase"/>

  <owl:Class rdf:ID="Category">
    <rdfs:label>Category</rdfs:label>
    <rdfs:comment xml:lang="en">
      Categories differ from Topics in that only one super category is allowed.
      Categories are not meant to be constructed as such but to be projected
      from an ontology based knowledge base e.g. Topic-definitions made with Amaya
      or FMS Ontology.

      A category definition should have rdfs:label and possibly also rdfs:comment.
      A category may also have weight specified, which should be interpreted as weight
      of a triple _:c1 ogt:subCategoryOf _:c2, where _:c1 ogt:weight X.
      This interpretation is unambiguous since categories can have at most one super category.
    </rdfs:comment>
    <rdfs:subClassOf rdf:resource="&bm;Topic"/>
  </owl:Class>

  <ogt:Category rdf:ID="RootCategory">
    <rdfs:comment xml:lang="en">
      Top level categories should be sub categories of ogt:RootCategory.
    </rdfs:comment>
  </ogt:Category>

  <owl:ObjectProperty rdf:ID="catid">
    <rdfs:comment xml:lang="en">
      Implementation dependent identifier for category.
    </rdfs:comment>
    <rdfs:domain rdf:resource="#Category"/>
    <rdfs:range rdf:resource="&xsd:string"/>
  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="projectionOf">
    <rdfs:comment xml:lang="en">
      If the category is a projection of some (domain ontology) resource
      the original resource is identified using this property.
    </rdfs:comment>
    <rdfs:domain rdf:resource="#Category"/>
    <rdfs:range rdf:resource="&rdfs;Resource"/>
  </owl:ObjectProperty>
```

```

<owl:FunctionalProperty rdf:ID="subCategoryOf">
  <rdfs:comment xml:lang="en">
    Super category.
    Only one super category allowed per Category.
  </rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&bm;subTopicOf"/>
  <rdfs:domain rdf:resource="#Category"/>
  <rdfs:range rdf:resource="#Category"/>
</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="hasRoot">
  <rdfs:comment xml:lang="en">
    The root of the category hierarchy.
    Only one root category allowed per Category.
  </rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&bm;subTopicOf"/>
  <rdfs:domain rdf:resource="#Category"/>
  <rdfs:range rdf:resource="#Category"/>
</owl:FunctionalProperty>

<owl:ObjectProperty rdf:ID="subCategories">
  <rdfs:comment xml:lang="en">
    Sub categories (ordered list).
    Values of this property are collections.
  </rdfs:comment>
  <owl:inverseOf rdf:resource="#subCategoryOf"/>
  <rdfs:domain rdf:resource="#Category"/>
  <rdfs:range rdf:resource="#Category"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="bookmarkHits">
  <rdfs:comment xml:lang="en">
    Hits provides an estimation of the size of the result set
    with current restrictions.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Category"/>
  <rdfs:range rdf:resource="&xsd;nonNegativeInteger"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="directBookmarkHits">
  <rdfs:comment xml:lang="en">
    Number of hits related directly to this category.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Category"/>
  <rdfs:range rdf:resource="&xsd;nonNegativeInteger"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="subCategoryHits">
  <rdfs:comment xml:lang="en">
    Number of unfiltered direct sub categories.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Category"/>
  <rdfs:range rdf:resource="&xsd;nonNegativeInteger"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="topicOf">
  <rdfs:comment xml:lang="en">
    TopicOf-property is used to define bookmarks that
    the category is topic of.
  </rdfs:comment>
  <owl:inverseOf rdf:resource="&bm;hasTopic"/>
  <rdfs:domain rdf:resource="&bm;Topic"/>
  <rdfs:range rdf:resource="&bm;Bookmark"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="Facet">
  <rdfs:comment xml:lang="en">

```

```

    Results for FacetSelector.
  </rdfs:comment>
  <rdfs:label xml:lang="en">Facet</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Category"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="remainder">
  <rdfs:comment xml:lang="en">
    Ungrouped bookmark hits.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Facet"/>
  <rdfs:range rdf:resource="&xsd:nonNegativeInteger"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="facetSelector">
  <rdfs:comment></rdfs:comment>
  <rdfs:domain rdf:resource="#Facet"/>
  <rdfs:range rdf:resource="#FacetSelector"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="CategoryConstraint">
  <rdfs:comment xml:lang="en">
    CategoryConstraint is used to express category-based constraints
    and to select categories.
  </rdfs:comment>
</owl:Class>

<owl:ObjectProperty rdf:ID="category">
  <rdfs:comment>
    Select or constraint category by its ogt:catid.
    If no category is provided the selector
    matches all (top) categories. If an instance specifies
    multiple values for ogt:category, the constraint is
    union of these.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#CategoryConstraint"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="projectedFrom">
  <rdfs:comment>
    Constraint categories by of which resource they are projected from.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#CategoryConstraint"/>
  <rdfs:range rdf:resource="&rdfs;Resource"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="categoryKeyword">
  <rdfs:comment>
    Select or constraint categories by keyword.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#CategoryConstraint"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="BookmarkConstraint">
  <rdfs:comment xml:lang="en">
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#CategoryConstraint"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="bookmarkKeyword">
  <rdfs:comment>
    Select or constraint bookmarks by keyword.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#BookmarkConstraint"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:ID="bookmark">
  <rdfs:comment></rdfs:comment>
  <rdfs:domain rdf:resource="#BookmarkConstraint"/>
  <rdfs:range rdf:resource="&rdfs;Resource"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="FacetSelector">
  <rdfs:label>FacetSelector</rdfs:label>
  <rdfs:comment>
    Query for Ontogator.
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#BookmarkConstraint"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="constraint">
  <rdfs:comment xml:lang="en">
    Constraint-property is used to express bookmark-constraints of a
    view-based query. If multiple constraints are given the result is
    the intersection of these. Constraints are given as rdf:List
    e.g. using rdf:parseType="Collection".
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="#BookmarkConstraint"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="acceptLanguage">
  <rdfs:comment xml:lang="en">
    Accepted xml:lang for literals (e.g. rdfs:label).
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="filter">
  <rdfs:comment xml:lang="en">
    Filtering type:

    1) NONE - no filtering,
    2) PRE-QUERY - categories that have no bookmarks are filtered
    3) POST-QUERY - categories with no bookmark hits are filtered
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="incBookmarkHits">
  <rdfs:comment xml:lang="en">
    Should the number of bookmark hits (ogt:bookmarkHits) be included
    in the result?
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="incPathBookmarkHits">
  <rdfs:comment xml:lang="en">
    Should the number of bookmark hits (ogt:bookmarkHits) be included
    in the path (ogt:hasRoot) of the result categories?
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="incDirectBookmarkHits">
  <rdfs:comment xml:lang="en">
    Should the number of direct bookmark hits (ogt:directBookmarkHits)
    be included in the result?
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

```



```

</rdfs:comment>
<rdfs:domain rdf:resource="#FacetSelector"/>
<rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="incRemainder">
  <rdfs:comment xml:lang="en">
    Should the number of ungrouped bookmark hits (ogt:remainder) be
    included in the result?
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="incPosition">
  <rdfs:comment xml:lang="en">
    Should the position of sub categories and bookmarks (ogt:position)
    be included in the result?
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="incSubCategoryHits">
  <rdfs:comment xml:lang="en">
    Should the number of sub category hits (ogt:subCategoryHits) be
    included in the result?
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="incPathSubCategoryHits">
  <rdfs:comment xml:lang="en">
    Should the number of sub category hits (ogt:subCategoryHits)
    be included in the path (ogt:hasRoot) of the result categories?
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="incCategoryProperty">
  <rdfs:comment xml:lang="en">
    Property of a category to be included in the results.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&rdf:Property"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="incBookmarkProperty">
  <rdfs:comment xml:lang="en">
    Property of a bookmark to be included in the results.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&rdfs:Property"/>
</owl:ObjectProperty>

<rdf:Property rdf:ID="ANY_PROPERTY">
  <rdfs:comment xml:lang="en">
    Selector that matches all properties.
    To be used with incCategoryProperty and incBookmarkProperty.
  </rdfs:comment>
</rdf:Property>

<owl:DatatypeProperty rdf:ID="incBookmarks">
  <rdfs:comment xml:lang="en">
    Should URIs of bookmarks related to a category
    be included in the category hierarchy?

```

```

    Default: false
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="incPath">
  <rdfs:comment xml:lang="en">
    Should parents (ogt:hasRoot) of the selected category
    be included in the category hierarchy?
    In a sense this is the counter part to the
    ogt:incSubCategories property.

    Default: true
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="incTopics">
  <rdfs:comment xml:lang="en">
    Should topics of categories be included in the result
    (Bookmark#hasTopic):

    1) NONE - no topics (default),
    2) DIRECT - direct topics,
    3) ALL - direct topics with paths (ogt:hasRoot).
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="incSubCategories">
  <rdfs:comment xml:lang="en">
    What sub categories, ALL, DIRECT or NONE,
    should be included in the result?

    Default: DIRECT
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="maxBookmarks">
  <rdfs:comment xml:lang="en">
    Max bookmarks of a given category to be included in the result.
    This restriction is inherited, so if ogt:incSubCategories=ALL and
    ogt:maxBookmarks [ogt:limit = 4], no more than 4 bookmarks will be
    included in the result for each category in the specified hierarchy.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="#Restriction"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="maxSubCategories">
  <rdfs:comment xml:lang="en">
    Max sub categories of a given category to be included in the result.
    This restriction is inherited, so if ogt:incSubCategories=ALL and
    ogt:maxBookmarks [ogt:limit = "4"], no more than 4 sub categories
    will be included in the result for each category in the specified
    hierarchy.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="#Restriction"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="skipBookmarks">

```

```

<rdfs:comment xml:lang="en">
  Number of bookmarks of the given category to be skipped.
</rdfs:comment>
<rdfs:domain rdf:resource="#FacetSelector"/>
<rdfs:range rdf:resource="#Restriction"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="skipSubCategories">
  <rdfs:comment xml:lang="en">
    Number of sub categories of the given category to be skipped.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#FacetSelector"/>
  <rdfs:range rdf:resource="#Restriction"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="Restriction">
  <rdfs:comment xml:lang="en">
    Restriction specifies the target category and the limit.
    If no category is specified the target is the resulting Facet.
  </rdfs:comment>
</owl:Class>

<owl:DatatypeProperty rdf:ID="onCategory">
  <rdfs:comment xml:lang="en">
    Specifies the target category (ogt:catid) of a restriction.
    If no category is specified the target is the resulting Facet.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Restriction"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="limit">
  <rdfs:comment xml:lang="en">
    Specifies the limit of a restriction.
    Negative values indicate no limit.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Restriction"/>
  <rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>
</rdf:RDF>

```

C Esimerkki kyselystä Ontogatorille

Esimerkkikysely Ontogatorille. Kysely on MuseoSuomi-järjestelmän automaattisesti tuottama. Kohteiden hakuehtoina käytetään kategorioita `%00%0C%0F` ja hakusanaa " espoo ". Vastaus tähän kyselyyn liitteessä D.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema#"
  xmlns:ui="http://www.cs.helsinki.fi/group/seco/ns/2004/03/ustate#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:fms="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#"
  xmlns:ogt="http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#"
  xmlns:bm="http://www.w3.org/2002/01/bookmark#"
  xmlns="http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#">
  <ogt:FacetSelector ogt:incSubCategories="NONE"
    ogt:category="%00%0C%0F"
    ogt:acceptLanguage="fi"
    ogt:incTopics="false"
    ogt:incBookmarkHits="false"
    ogt:incCategories="true"
    ogt:incBookmarks="false"
    ogt:incPath="true"
    ogt:filter="NONE"
    rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F">
    <ogt:incPath>true</ogt:incPath>
    <ogt:incCategoryProperty
      rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
  </ogt:FacetSelector>
  <ogt:FacetSelector ogt:incSubCategoryHits="true"
    ogt:incTopics="NONE"
    ogt:incSubCategories="DIRECT"
    ogt:incCategories="true"
    ogt:incBookmarks="false"
    ogt:acceptLanguage="fi"
    ogt:filter="POST-QUERY"
    ogt:incPath="true"
    rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%00%0C%0F">
    <ogt:maxSubCategories ogt:limit="20"/>
    <ogt:category>%00%0C%0F</ogt:category>
    <ogt:incCategoryProperty
      rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
    <ogt:constraint rdf:parseType="Collection">
      <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
      <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
    </ogt:constraint>
  </ogt:FacetSelector>
  <ogt:FacetSelector ogt:incSubCategoryHits="true"
    ogt:incTopics="NONE"
    ogt:incSubCategories="DIRECT"
    ogt:incCategories="true"
    ogt:incBookmarks="false"
    ogt:acceptLanguage="fi"
    ogt:filter="POST-QUERY"
    ogt:incPath="true"
    rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%01">
    <ogt:maxSubCategories ogt:limit="20"/>
    <ogt:category>%01</ogt:category>
    <ogt:incCategoryProperty
      rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
    <ogt:constraint rdf:parseType="Collection">
      <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
      <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
    </ogt:constraint>
  </ogt:FacetSelector>
  <ogt:FacetSelector ogt:incSubCategoryHits="true"
    ogt:incTopics="NONE"
    ogt:incSubCategories="DIRECT"
```

```

        ogt:incCategories="true"
        ogt:incBookmarks="false"
        ogt:acceptLanguage="fi"
        ogt:filter="POST-QUERY"
        ogt:incPath="true"
        rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%02">
<ogt:maxSubCategories ogt:limit="20"/>
<ogt:category>%02</ogt:category>
<ogt:incCategoryProperty
  rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
<ogt:constraint rdf:parseType="Collection">
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
</ogt:constraint>
</ogt:FacetSelector>
<ogt:FacetSelector ogt:incSubCategoryHits="true"
  ogt:incTopics="NONE"
  ogt:incSubCategories="DIRECT"
  ogt:incCategories="true"
  ogt:incBookmarks="false"
  ogt:acceptLanguage="fi"
  ogt:filter="POST-QUERY"
  ogt:incPath="true"
  rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%03">
<ogt:maxSubCategories ogt:limit="20"/>
<ogt:category>%03</ogt:category>
<ogt:incCategoryProperty
  rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
<ogt:constraint rdf:parseType="Collection">
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
</ogt:constraint>
</ogt:FacetSelector>
<ogt:FacetSelector ogt:incSubCategoryHits="true"
  ogt:incTopics="NONE"
  ogt:incSubCategories="DIRECT"
  ogt:incCategories="true"
  ogt:incBookmarks="false"
  ogt:acceptLanguage="fi"
  ogt:filter="POST-QUERY"
  ogt:incPath="true"
  rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%04">
<ogt:maxSubCategories ogt:limit="20"/>
<ogt:category>%04</ogt:category>
<ogt:incCategoryProperty
  rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
<ogt:constraint rdf:parseType="Collection">
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
</ogt:constraint>
</ogt:FacetSelector>
<ogt:FacetSelector ogt:incSubCategoryHits="true"
  ogt:incTopics="NONE"
  ogt:incSubCategories="DIRECT"
  ogt:incCategories="true"
  ogt:incBookmarks="false"
  ogt:acceptLanguage="fi"
  ogt:filter="POST-QUERY"
  ogt:incPath="true"
  rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%05">
<ogt:maxSubCategories ogt:limit="20"/>
<ogt:category>%05</ogt:category>
<ogt:incCategoryProperty
  rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
<ogt:constraint rdf:parseType="Collection">
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
</ogt:constraint>

```

```

</ogt:FacetSelector>
<ogt:FacetSelector ogt:incSubCategoryHits="true"
  ogt:incTopics="NONE"
  ogt:incSubCategories="DIRECT"
  ogt:incCategories="true"
  ogt:incBookmarks="false"
  ogt:acceptLanguage="fi"
  ogt:filter="POST-QUERY"
  ogt:incPath="true"
  rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%06">
  <ogt:maxSubCategories ogt:limit="20"/>
  <ogt:category>%06</ogt:category>
  <ogt:incCategoryProperty
    rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
  <ogt:constraint rdf:parseType="Collection">
    <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
    <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
  </ogt:constraint>
</ogt:FacetSelector>
<ogt:FacetSelector ogt:incSubCategoryHits="true"
  ogt:incTopics="NONE"
  ogt:incSubCategories="DIRECT"
  ogt:incCategories="true"
  ogt:incBookmarks="false"
  ogt:acceptLanguage="fi"
  ogt:filter="POST-QUERY"
  ogt:incPath="true"
  rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%07">
  <ogt:maxSubCategories ogt:limit="20"/>
  <ogt:category>%07</ogt:category>
  <ogt:incCategoryProperty
    rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
  <ogt:constraint rdf:parseType="Collection">
    <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
    <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
  </ogt:constraint>
</ogt:FacetSelector>
<ogt:FacetSelector ogt:incSubCategoryHits="true"
  ogt:incTopics="NONE"
  ogt:incSubCategories="DIRECT"
  ogt:incCategories="true"
  ogt:incBookmarks="false"
  ogt:acceptLanguage="fi"
  ogt:filter="POST-QUERY"
  ogt:incPath="true"
  rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%08">
  <ogt:maxSubCategories ogt:limit="20"/>
  <ogt:category>%08</ogt:category>
  <ogt:incCategoryProperty
    rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
  <ogt:constraint rdf:parseType="Collection">
    <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
    <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
  </ogt:constraint>
</ogt:FacetSelector>
<ogt:FacetSelector ogt:incSubCategoryHits="true"
  ogt:incSubCategories="NONE"
  ogt:incCategories="true"
  ogt:incBookmarks="false"
  ogt:acceptLanguage="fi"
  ogt:filter="POST-QUERY"
  rdfs:label="espoo"
  ogt:incPath="true"
  rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo">
  <ogt:maxBookmarks ogt:limit="4"/>
  <ogt:maxSubCategories ogt:limit="20"/>
  <ogt:categoryKeyword>espoo</ogt:categoryKeyword>
  <ogt:bookmarkKeyword>espoo</ogt:bookmarkKeyword>

```

```

<ogt:incCategoryProperty
  rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
<ogt:constraint rdf:parseType="Collection">
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
  <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
  </ogt:constraint>
</ogt:FacetSelector>
<ogt:FacetSelector rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#bookmarksByCategory">
  <ogt:incCategoryProperty
    rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#rootcatid"/>
  <ogt:incSubCategories>DIRECT</ogt:incSubCategories>
  <ogt:category>%00%0C%0F</ogt:category>
  <ogt:incCategories>true</ogt:incCategories>
  <ogt:filter>POST-QUERY</ogt:filter>
  <ogt:incPath>true</ogt:incPath>
  <ogt:incTopics>NONE</ogt:incTopics>
  <ogt:incDirectBookmarkHits>true</ogt:incDirectBookmarkHits>
  <ogt:incRemainder>true</ogt:incRemainder>
  <ogt:maxBookmarks ogt:limit="4"/>
  <ogt:maxSubCategories ogt:limit="7"/>
  <ogt:incBookmarks>true</ogt:incBookmarks>
  <ogt:incSubCategoryHits>true</ogt:incSubCategoryHits>
  <ogt:acceptLanguage>fi</ogt:acceptLanguage>
  <ogt:incBookmarkProperty
    rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#www_kuvatiedostonimi"/>
  <ogt:incBookmarkProperty
    rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#www_numero"/>
  <ogt:constraint rdf:parseType="Collection">
    <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
    <rdf:Description rdf:about="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"/>
  </ogt:constraint>
</ogt:FacetSelector>
</rdf:RDF>

```

D Esimerkki Ontogatorin vastauksesta

Ontogatorin vastaus kyselyyn, jossa käytetään hakuehtoina kategoriää `%00%0C%0F` ja hakusanaa "espo". Tämän vastauksen tuottanut kysely liitteessä C.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ogt="http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#"
  xmlns:fms="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:bm="http://www.w3.org/2002/01/bookmark#"
  xmlns:swhg="http://www.cs.helsinki.fi/group/seco/ontologies/ontodella/sewehgrius.rdfs#">
  <ogt:CategoryBase rdf:about="http://www.cs.helsinki.fi/group/seco/ontologies/ontogator/vbsearch">
    <od:sewehgrius_dataid>silent_squirrel</od:sewehgrius_dataid>
    <owl:versionInfo>silent_squirrel</owl:versionInfo>
    <od:ontodellaHost>pehesaari-18.cs.helsinki.fi</od:ontodellaHost>
    <od:ontodellaCvsId>
      $Id: ontodella_request_handler.pl,v 1.41 2004/03/04 23:08:31 kimvilja Exp $
    </od:ontodellaCvsId>
    <od:sewehgrius_ruleid>
      $Id: fmsweb_sewehgrius_rules.pl,v 1.36 2004/03/05 19:03:26 kimvilja Exp $
    </od:sewehgrius_ruleid>    <od:replyCreationTime>Sun Mar 7 03:25:07 2004</od:replyCreationTime>
  </ogt:CategoryBase>
  <ogt:Facet>
    <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%04"/>
    <ogt:bookmarkHits>57</ogt:bookmarkHits>
    <ogt:subCategories rdf:parseType="Collection">
      <ogt:Category ogt:catid="%04" rdfs:label="Valmistusaika">
        <ogt:bookmarkHits>57</ogt:bookmarkHits>
        <fms:rootcatid>4</fms:rootcatid>
        <ogt:subCategories rdf:parseType="Collection">
          <ogt:Category ogt:catid="%04%00" rdfs:label="aikakaudet">
            <ogt:bookmarkHits>57</ogt:bookmarkHits>
            <ogt:subCategoryHits>1</ogt:subCategoryHits>
          </ogt:Category>
          <ogt:Category ogt:catid="%04%01" rdfs:label="vuosisadat">
            <ogt:bookmarkHits>57</ogt:bookmarkHits>
            <ogt:subCategoryHits>1</ogt:subCategoryHits>
          </ogt:Category>
        </ogt:subCategories>
        <ogt:subCategoryHits>2</ogt:subCategoryHits>
      </ogt:Category>
    </ogt:subCategories>
    <ogt:subCategoryHits>1</ogt:subCategoryHits>
  </ogt:Facet>
  <ogt:Facet>
    <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%03"/>
    <ogt:bookmarkHits>57</ogt:bookmarkHits>
    <ogt:subCategories rdf:parseType="Collection">
      <ogt:Category ogt:catid="%03" rdfs:label="Valmistuspaikka">
        <ogt:bookmarkHits>51</ogt:bookmarkHits>
        <fms:rootcatid>3</fms:rootcatid>
        <ogt:subCategories rdf:parseType="Collection">
          <ogt:Category ogt:catid="%03%03" rdfs:label="Eurooppa">
            <ogt:bookmarkHits>51</ogt:bookmarkHits>
            <ogt:subCategoryHits>3</ogt:subCategoryHits>
          </ogt:Category>
        </ogt:subCategories>
        <ogt:subCategoryHits>1</ogt:subCategoryHits>
      </ogt:Category>
    </ogt:subCategories>
    <ogt:subCategoryHits>1</ogt:subCategoryHits>
  </ogt:Facet>
  <ogt:Facet>
    <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%02"/>
    <ogt:bookmarkHits>57</ogt:bookmarkHits>
    <ogt:subCategories rdf:parseType="Collection">
```



```

<ogt:Category ogt:catid="%02" rdfs:label="Valmistaja">
  <ogt:bookmarkHits>34</ogt:bookmarkHits>
  <fms:rootcatid>2</fms:rootcatid>
  <ogt:subCategories rdf:parseType="Collection">
    <ogt:Category ogt:catid="%02%00" rdfs:label="henkilöt">
      <ogt:bookmarkHits>1</ogt:bookmarkHits>
      <ogt:subCategoryHits>1</ogt:subCategoryHits>
    </ogt:Category>
    <ogt:Category ogt:catid="%02%06" rdfs:label="yhteisöt">
      <ogt:bookmarkHits>1</ogt:bookmarkHits>
      <ogt:subCategoryHits>1</ogt:subCategoryHits>
    </ogt:Category>
    <ogt:Category ogt:catid="%02%07" rdfs:label="yritykset">
      <ogt:bookmarkHits>34</ogt:bookmarkHits>
      <ogt:subCategoryHits>18</ogt:subCategoryHits>
    </ogt:Category>
  </ogt:subCategories>
  <ogt:subCategoryHits>3</ogt:subCategoryHits>
</ogt:Category>
</ogt:subCategories>
<ogt:subCategoryHits>1</ogt:subCategoryHits>
</ogt:Facet>
<ogt:Facet>
  <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%01"/>
  <ogt:bookmarkHits>57</ogt:bookmarkHits>
  <ogt:subCategories rdf:parseType="Collection">
    <ogt:Category ogt:catid="%01" rdfs:label="Materiaali">
      <ogt:bookmarkHits>43</ogt:bookmarkHits>
      <fms:rootcatid>1</fms:rootcatid>
      <ogt:subCategories rdf:parseType="Collection">
        <ogt:Category ogt:catid="%01%03" rdfs:label="materiaalit">
          <ogt:bookmarkHits>43</ogt:bookmarkHits>
          <ogt:subCategoryHits>4</ogt:subCategoryHits>
        </ogt:Category>
      </ogt:subCategories>
      <ogt:subCategoryHits>1</ogt:subCategoryHits>
    </ogt:Category>
  </ogt:subCategories>
  <ogt:subCategoryHits>1</ogt:subCategoryHits>
</ogt:Facet>
<ogt:Facet>
  <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#kespoo"
    rdfs:label="espool"/>
  <ogt:bookmarkHits>57</ogt:bookmarkHits>
  <ogt:subCategories rdf:parseType="Collection">
    <ogt:Category ogt:catid="%03%03%0E%00%05%00" rdfs:label="Espoo">
      <ogt:bookmarkHits>13</ogt:bookmarkHits>
      <ogt:subCategoryOf ogt:catid="%03%03%0E%00%05" rdfs:label="Uusimaa-Nyland"/>
      <ogt:hasRoot>
        <ogt:Category ogt:catid="%03" rdfs:label="Valmistuspaikka">
          <fms:rootcatid>3</fms:rootcatid>
          <ogt:subCategories rdf:parseType="Collection">
            <ogt:Category ogt:catid="%03%03" rdfs:label="Eurooppa">
              <ogt:subCategories rdf:parseType="Collection">
                <ogt:Category ogt:catid="%03%03%0E" rdfs:label="Suomi">
                  <ogt:subCategories rdf:parseType="Collection">
                    <ogt:Category ogt:catid="%03%03%0E%00" rdfs:label="Etelä-Suomen lääni">
                      <ogt:subCategories rdf:parseType="Collection">
                        <ogt:Category ogt:catid="%03%03%0E%00%05" rdfs:label="Uusimaa-Nyland"/>
                      </ogt:subCategories>
                    </ogt:Category>
                  </ogt:subCategories>
                </ogt:Category>
              </ogt:subCategories>
            </ogt:Category>
          </ogt:subCategories>
        </ogt:Category>
      </ogt:hasRoot>
    </ogt:Category>
  </ogt:subCategories>
  <ogt:subCategoryHits>1</ogt:subCategoryHits>
</ogt:Facet>
</ogt:Facet>

```

```

    <ogt:subCategoryHits>2</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%06%03%0E%00%05%00" rdfs:label="Espoo">
    <ogt:bookmarkHits>16</ogt:bookmarkHits>
    <ogt:subCategoryOf ogt:catid="%06%03%0E%00%05" rdfs:label="Uusimaa-Nyland"/>
    <ogt:hasRoot>
      <ogt:Category ogt:catid="%06" rdfs:label="Käyttöpaikka">
        <fms:rootcatid>6</fms:rootcatid>
        <ogt:subCategories rdf:parseType="Collection">
          <ogt:Category ogt:catid="%06%03" rdfs:label="Eurooppa">
            <ogt:subCategories rdf:parseType="Collection">
              <ogt:Category ogt:catid="%06%03%0E" rdfs:label="Suomi">
                <ogt:subCategories rdf:parseType="Collection">
                  <ogt:Category ogt:catid="%06%03%0E%00" rdfs:label="Etelä-Suomen lääni">
                    <ogt:subCategories rdf:parseType="Collection">
                      <ogt:Category ogt:catid="%06%03%0E%00%05" rdfs:label="Uusimaa-Nyland"/>
                    </ogt:subCategories>
                  </ogt:Category>
                </ogt:subCategories>
              </ogt:Category>
            </ogt:subCategories>
          </ogt:Category>
        </ogt:subCategories>
      </ogt:hasRoot>
      <ogt:subCategoryHits>4</ogt:subCategoryHits>
    </ogt:Category>
    <ogt:Category ogt:catid="%08%00" rdfs:label="Espoon kaupunginmuseon kokoelmat">
      <ogt:bookmarkHits>56</ogt:bookmarkHits>
      <ogt:subCategoryOf ogt:catid="%08" rdfs:label="Kokoelma"/>
      <ogt:hasRoot>
        <ogt:Category ogt:catid="%08" rdfs:label="Kokoelma">
          <fms:rootcatid>8</fms:rootcatid>
        </ogt:Category>
      </ogt:hasRoot>
      <ogt:subCategoryHits>1</ogt:subCategoryHits>
    </ogt:Category>
  </ogt:subCategories>
  <ogt:subCategoryHits>3</ogt:subCategoryHits>
</ogt:Facet>
<ogt:Facet>
  <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#s%00%0C%0F"/>
  <ogt:subCategories rdf:parseType="Collection">
    <ogt:Category ogt:catid="%00%0C%0F" rdfs:label="säilytysastiat">
      <ogt:subCategoryOf ogt:catid="%00%0C" rdfs:label="säilyttimet"/>
      <ogt:hasRoot>
        <ogt:Category ogt:catid="%00" rdfs:label="Esinetyyppi">
          <fms:rootcatid>0</fms:rootcatid>
          <ogt:subCategories rdf:parseType="Collection">
            <ogt:Category ogt:catid="%00%0C" rdfs:label="säilyttimet"/>
          </ogt:subCategories>
        </ogt:Category>
      </ogt:hasRoot>
    </ogt:Category>
  </ogt:subCategories>
</ogt:Facet>
<ogt:Facet>
  <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#bookmarksByCategory"/>
  <ogt:bookmarkHits>57</ogt:bookmarkHits>
  <ogt:subCategories rdf:parseType="Collection">
    <ogt:Category ogt:catid="%00%0C%0F" rdfs:label="säilytysastiat">
      <ogt:bookmarkHits>57</ogt:bookmarkHits>
      <ogt:subCategoryOf ogt:catid="%00%0C" rdfs:label="säilyttimet"/>
      <ogt:hasRoot>
        <ogt:Category ogt:catid="%00" rdfs:label="Esinetyyppi">
          <fms:rootcatid>0</fms:rootcatid>
          <ogt:subCategories rdf:parseType="Collection">
            <ogt:Category ogt:catid="%00%0C" rdfs:label="säilyttimet"/>
          </ogt:subCategories>
        </ogt:Category>
      </ogt:hasRoot>
    </ogt:Category>
  </ogt:subCategories>
</ogt:Facet>

```

```

    </ogt:subCategories>
  </ogt:Category>
</ogt:hasRoot>
<ogt:subCategories rdf:parseType="Collection">
  <ogt:Category ogt:catid="%00%0C%0F%01" rdfs:label="kanisterit">
    <ogt:bookmarkHits>7</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
    <ogt:topicOf rdf:parseType="Collection">
      <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2194_66"
        rdfs:label="kanisteri:taskumatti">
        <fms:www_numero>ECM:2194:66</fms:www_numero>
        <fms:www_kuvatiedostonimi>
          /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000581.png
        </fms:www_kuvatiedostonimi>
      </bm:Bookmark>
      <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2199_168"
        rdfs:label="kanisteri:taskumatti">
        <fms:www_numero>ECM:2199:168</fms:www_numero>
        <fms:www_kuvatiedostonimi>
          /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000580.png
        </fms:www_kuvatiedostonimi>
      </bm:Bookmark>
      <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2199_169"
        rdfs:label="kanisteri:taskumatti">
        <fms:www_numero>ECM:2199:169</fms:www_numero>
        <fms:www_kuvatiedostonimi>
          /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000580.png
        </fms:www_kuvatiedostonimi>
      </bm:Bookmark>
      <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2423_253"
        rdfs:label="kanisteri:taskumatti">
        <fms:www_numero>ECM:2423:253</fms:www_numero>
        <fms:www_kuvatiedostonimi>
          /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000559.png
        </fms:www_kuvatiedostonimi>
      </bm:Bookmark>
    </ogt:topicOf>
  </ogt:Category>
<ogt:Category ogt:catid="%00%0C%0F%02" rdfs:label="maljakot">
  <ogt:bookmarkHits>11</ogt:bookmarkHits>
  <ogt:subCategoryHits>0</ogt:subCategoryHits>
  <ogt:topicOf rdf:parseType="Collection">
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2127_1"
      rdfs:label="maljakko:maljakko">
      <fms:www_numero>ECM:2127:1</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000599.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2286_4"
      rdfs:label="maljakko:maljakko">
      <fms:www_numero>ECM:2286:4</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000611.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_3318_1"
      rdfs:label="maljakko">
      <fms:www_numero>ECM:3318:1</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000006.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_3443_1"
      rdfs:label="maljakko:maljakko">
      <fms:www_numero>ECM:3443:1</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0001004.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
  </ogt:topicOf>
</ogt:Category>

```

```

        </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
</ogt:topicOf>
</ogt:Category>
<ogt:Category ogt:catid="%00%0C%0F%04" rdfs:label="pakit">
    <ogt:bookmarkHits>1</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
    <ogt:topicOf rdf:parseType="Collection">
        <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_3594_117"
            rdfs:label="ruokapakki:kenttäkeittoastia">
            <fms:www_numero>ECM:3594:117</fms:www_numero>
            <fms:www_kuvatiedostonimi>
                /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0001896.png
            </fms:www_kuvatiedostonimi>
        </bm:Bookmark>
    </ogt:topicOf>
</ogt:Category>
<ogt:Category ogt:catid="%00%0C%0F%05" rdfs:label="purkit">
    <ogt:bookmarkHits>16</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
    <ogt:topicOf rdf:parseType="Collection">
        <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2343_14"
            rdfs:label="hiivatölkki:panimohiivapurkki">
            <fms:www_numero>ECM:2343:14</fms:www_numero>
            <fms:www_kuvatiedostonimi>
                /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0001350.png
            </fms:www_kuvatiedostonimi>
        </bm:Bookmark>
        <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2498_78"
            rdfs:label="kahvitölkki:kahvipurkki">
            <fms:www_numero>ECM:2498:78</fms:www_numero>
            <fms:www_kuvatiedostonimi>
                /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000442.png
            </fms:www_kuvatiedostonimi>
        </bm:Bookmark>
        <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2669_31"
            rdfs:label="maitotölkki:maitopurkki">
            <fms:www_numero>ECM:2669:31</fms:www_numero>
            <fms:www_kuvatiedostonimi>
                /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0002006.png
            </fms:www_kuvatiedostonimi>
        </bm:Bookmark>
        <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2669_32"
            rdfs:label="maitotölkki:maitopurkki">
            <fms:www_numero>ECM:2669:32</fms:www_numero>
            <fms:www_kuvatiedostonimi>
                /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0002006.png
            </fms:www_kuvatiedostonimi>
        </bm:Bookmark>
    </ogt:topicOf>
</ogt:Category>
<ogt:Category ogt:catid="%00%0C%0F%06" rdfs:label="ruukut">
    <ogt:bookmarkHits>2</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
    <ogt:topicOf rdf:parseType="Collection">
        <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_3572_24"
            rdfs:label="ruukku:talousruukku">
            <fms:www_numero>ECM:3572:24</fms:www_numero>
            <fms:www_kuvatiedostonimi>
                /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0001487.png
            </fms:www_kuvatiedostonimi>
        </bm:Bookmark>
        <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_3579_4"
            rdfs:label="ruukku:karjalanruukku">
            <fms:www_numero>ECM:3579:4</fms:www_numero>
            <fms:www_kuvatiedostonimi>
                /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0001543.png
            </fms:www_kuvatiedostonimi>
    </ogt:topicOf>
</ogt:Category>

```

```

    </bm:Bookmark>
  </ogt:topicOf>
</ogt:Category>
<ogt:Category ogt:catid="%00%0C%0F%08" rdfs:label="taskumatit">
  <ogt:bookmarkHits>6</ogt:bookmarkHits>
  <ogt:subCategoryHits>0</ogt:subCategoryHits>
  <ogt:topicOf rdf:parseType="Collection">
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2194_66"
      rdfs:label="kanisteri:taskumatti">
      <fms:www_numero>ECM:2194:66</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000581.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2199_168"
      rdfs:label="kanisteri:taskumatti">
      <fms:www_numero>ECM:2199:168</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000580.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2199_169"
      rdfs:label="kanisteri:taskumatti">
      <fms:www_numero>ECM:2199:169</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000580.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2423_253"
      rdfs:label="kanisteri:taskumatti">
      <fms:www_numero>ECM:2423:253</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000559.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
  </ogt:topicOf>
</ogt:Category>
<ogt:Category ogt:catid="%00%0C%0F%09" rdfs:label="tölkit">
  <ogt:bookmarkHits>26</ogt:bookmarkHits>
  <ogt:subCategoryHits>0</ogt:subCategoryHits>
  <ogt:topicOf rdf:parseType="Collection">
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2343_14"
      rdfs:label="hiivatölkki:panimohiivapurkki">
      <fms:www_numero>ECM:2343:14</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0001350.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2372_4"
      rdfs:label="oluttölkki">
      <fms:www_numero>ECM:2372:4</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0001344.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2498_78"
      rdfs:label="kahvitölkki:kahvipurkki">
      <fms:www_numero>ECM:2498:78</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000442.png
      </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2669_31"
      rdfs:label="maitotölkki:maitopurkki">
      <fms:www_numero>ECM:2669:31</fms:www_numero>
      <fms:www_kuvatiedostonimi>
        /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e000/e0002006.png
      </fms:www_kuvatiedostonimi>
  </ogt:topicOf>
</ogt:Category>

```

```

        </bm:Bookmark>
    </ogt:topicOf>
</ogt:Category>
</ogt:subCategories>
<ogt:subCategoryHits>8</ogt:subCategoryHits>
<ogt:topicOf rdf:parseType="Collection">
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2037_37"
        rdfs:label="lamppu:umpikupu">
        <fms:www_numero>ECM:2037:37</fms:www_numero>
        <fms:www_kuvatiedostonimi>
            /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000613.png
        </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
    <bm:Bookmark rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_2978_3"
        rdfs:label="lamppu:umpikupu">
        <fms:www_numero>ECM:2978:3</fms:www_numero>
        <fms:www_kuvatiedostonimi>
            /home/group/seco/public_html/intra/Museo/EKM/Kuvat/e0000/e0000613.png
        </fms:www_kuvatiedostonimi>
    </bm:Bookmark>
</ogt:topicOf>
<ogt:directBookmarkHits>2</ogt:directBookmarkHits>
</ogt:Category>
</ogt:subCategories>
<ogt:subCategoryHits>1</ogt:subCategoryHits>
<ogt:topicOf rdf:parseType="Collection"/>
<ogt:remainder>0</ogt:remainder>
</ogt:Facet>
<ogt:Facet>
    <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%08"/>
    <ogt:bookmarkHits>57</ogt:bookmarkHits>
    <ogt:subCategories rdf:parseType="Collection">
        <ogt:Category ogt:catid="%08" rdfs:label="Kokoelma">
            <ogt:bookmarkHits>57</ogt:bookmarkHits>
            <fms:rootcatid>8</fms:rootcatid>
            <ogt:subCategories rdf:parseType="Collection">
                <ogt:Category ogt:catid="%08%00" rdfs:label="Espoon kaupunginmuseon kokoelmat">
                    <ogt:bookmarkHits>56</ogt:bookmarkHits>
                    <ogt:subCategoryHits>1</ogt:subCategoryHits>
                </ogt:Category>
                <ogt:Category ogt:catid="%08%02" rdfs:label="Lahden kaupunginmuseon kokoelmat">
                    <ogt:bookmarkHits>1</ogt:bookmarkHits>
                    <ogt:subCategoryHits>1</ogt:subCategoryHits>
                </ogt:Category>
            </ogt:subCategories>
            <ogt:subCategoryHits>2</ogt:subCategoryHits>
        </ogt:Category>
    </ogt:subCategories>
    <ogt:subCategoryHits>1</ogt:subCategoryHits>
</ogt:Facet>
<ogt:Facet>
    <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%07"/>
    <ogt:bookmarkHits>57</ogt:bookmarkHits>
    <ogt:subCategories rdf:parseType="Collection">
        <ogt:Category ogt:catid="%07" rdfs:label="Käyttötilanne">
            <ogt:bookmarkHits>56</ogt:bookmarkHits>
            <fms:rootcatid>7</fms:rootcatid>
            <ogt:subCategories rdf:parseType="Collection">
                <ogt:Category ogt:catid="%07%03" rdfs:label="juhlat ja seremoniat">
                    <ogt:bookmarkHits>1</ogt:bookmarkHits>
                    <ogt:subCategoryHits>1</ogt:subCategoryHits>
                </ogt:Category>
                <ogt:Category ogt:catid="%07%04" rdfs:label="kohteelle tehtävät toimenpiteet">
                    <ogt:bookmarkHits>5</ogt:bookmarkHits>
                    <ogt:subCategoryHits>2</ogt:subCategoryHits>
                </ogt:Category>
                <ogt:Category ogt:catid="%07%08" rdfs:label="ruoan- ja juomanvalmistus">
                    <ogt:bookmarkHits>13</ogt:bookmarkHits>
            </ogt:subCategories>
        </ogt:Category>
    </ogt:subCategories>
    <ogt:subCategoryHits>1</ogt:subCategoryHits>
</ogt:Facet>

```

```

    <ogt:subCategoryHits>0</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%07%09" rdfs:label="sodat">
    <ogt:bookmarkHits>1</ogt:bookmarkHits>
    <ogt:subCategoryHits>1</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%07%0B" rdfs:label="elinkeinot">
    <ogt:bookmarkHits>7</ogt:bookmarkHits>
    <ogt:subCategoryHits>2</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%07%0E" rdfs:label="valmistustekniikat">
    <ogt:bookmarkHits>31</ogt:bookmarkHits>
    <ogt:subCategoryHits>2</ogt:subCategoryHits>
  </ogt:Category>
</ogt:subCategories>
<ogt:subCategoryHits>6</ogt:subCategoryHits>
</ogt:Category>
</ogt:subCategories>
<ogt:subCategoryHits>1</ogt:subCategoryHits>
</ogt:Facet>
<ogt:Facet>
  <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%06"/>
  <ogt:bookmarkHits>57</ogt:bookmarkHits>
  <ogt:subCategories rdf:parseType="Collection">
    <ogt:Category ogt:catid="%06" rdfs:label="Käyttöpaikka">
      <ogt:bookmarkHits>29</ogt:bookmarkHits>
      <fms:rootcatid>6</fms:rootcatid>
      <ogt:subCategories rdf:parseType="Collection">
        <ogt:Category ogt:catid="%06%03" rdfs:label="Eurooppa">
          <ogt:bookmarkHits>29</ogt:bookmarkHits>
          <ogt:subCategoryHits>2</ogt:subCategoryHits>
        </ogt:Category>
      </ogt:subCategories>
      <ogt:subCategoryHits>1</ogt:subCategoryHits>
    </ogt:Category>
  </ogt:subCategories>
  <ogt:subCategoryHits>1</ogt:subCategoryHits>
</ogt:Facet>
<ogt:Facet>
  <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%05"/>
  <ogt:bookmarkHits>57</ogt:bookmarkHits>
  <ogt:subCategories rdf:parseType="Collection">
    <ogt:Category ogt:catid="%05" rdfs:label="Käyttäjät">
      <ogt:bookmarkHits>15</ogt:bookmarkHits>
      <fms:rootcatid>5</fms:rootcatid>
      <ogt:subCategories rdf:parseType="Collection">
        <ogt:Category ogt:catid="%05%00" rdfs:label="henkilöt">
          <ogt:bookmarkHits>15</ogt:bookmarkHits>
          <ogt:subCategoryHits>2</ogt:subCategoryHits>
        </ogt:Category>
      </ogt:subCategories>
      <ogt:subCategoryHits>1</ogt:subCategoryHits>
    </ogt:Category>
  </ogt:subCategories>
  <ogt:subCategoryHits>1</ogt:subCategoryHits>
</ogt:Facet>
<ogt:Facet>
  <ogt:facetSelector rdf:resource="http://museosuomi.cs.helsinki.fi/internal/ogtQuery#c%00%0C%0F"/>
  <ogt:bookmarkHits>57</ogt:bookmarkHits>
  <ogt:subCategories rdf:parseType="Collection">
    <ogt:Category ogt:catid="%00%0C%0F" rdfs:label="säilytysastiat">
      <ogt:bookmarkHits>57</ogt:bookmarkHits>
      <ogt:subCategoryOf ogt:catid="%00%0C" rdfs:label="säilyttimet"/>
      <ogt:hasRoot>
        <ogt:Category ogt:catid="%00" rdfs:label="Esinetyyppi">
          <fms:rootcatid>0</fms:rootcatid>
          <ogt:subCategories rdf:parseType="Collection">
            <ogt:Category ogt:catid="%00%0C" rdfs:label="säilyttimet"/>

```

```

    </ogt:subCategories>
  </ogt:Category>
</ogt:hasRoot>
<ogt:subCategories rdf:parseType="Collection">
  <ogt:Category ogt:catid="%00%0C%0F%01" rdfs:label="kanisterit">
    <ogt:bookmarkHits>7</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%00%0C%0F%02" rdfs:label="maljakot">
    <ogt:bookmarkHits>11</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%00%0C%0F%04" rdfs:label="pakit">
    <ogt:bookmarkHits>1</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%00%0C%0F%05" rdfs:label="purkit">
    <ogt:bookmarkHits>16</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%00%0C%0F%06" rdfs:label="ruukut">
    <ogt:bookmarkHits>2</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%00%0C%0F%08" rdfs:label="taskumatit">
    <ogt:bookmarkHits>6</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%00%0C%0F%09" rdfs:label="tölkit">
    <ogt:bookmarkHits>26</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
  </ogt:Category>
  <ogt:Category ogt:catid="%00%0C%0F%0A" rdfs:label="tonkat">
    <ogt:bookmarkHits>1</ogt:bookmarkHits>
    <ogt:subCategoryHits>0</ogt:subCategoryHits>
  </ogt:Category>
</ogt:subCategories>
<ogt:subCategoryHits>8</ogt:subCategoryHits>
</ogt:Category>
</ogt:subCategories>
<ogt:subCategoryHits>1</ogt:subCategoryHits>
</ogt:Facet>
</rdf:RDF>

```