

Monilähteinen suosittelu semanttisessa webissä

Kim Viljanen

Helsinki 23.2.2006

Pro gradu -tutkielma

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

| | | | |
|--|--|--|--|
| Tiedekunta/Osasto — Fakultet/Sektion — Faculty | | Laitos — Institution — Department | |
| Matemaattis-luonnontieteellinen | | Tietojenkäsittelytieteen laitos | |
| Tekijä — Författare — Author Kim Viljanen | | | |
| Työn nimi — Arbetets titel — Title Monilähteinen suosittelu semanttisessa webissä | | | |
| Oppiaine — Läroämne — Subject Tietojenkäsittelytiede | | | |
| Työn laji — Arbetets art — Level Pro gradu -tutkielma | | Aika — Datum — Month and year 23.2.2006 | Sivumäärä — Sidoantal — Number of pages 76 sivua + 4 liitesivua |
| Tiivistelmä — Referat — Abstract <p>Kuluttajan arkipäiväinen ongelma on löytää omiin tarpeisiinsa parhaiten sopivat tuotteet ja palvelut lukuisten vaihtoehtojen joukosta, joita on saatavilla sekä sähköisessä että perinteisessä muodossa. Esimerkiksi World Wide Webissä (WWW) on nykyisin yli 11,5 miljardia sivua. Tyypillisissä arkipäiväisissä valintatilanteissa kuluttaja ei pysty perehtymään kaikkiin tarjolla oleviin vaihtoehtoihin, joten tarvitaan apuvälineitä, joilla helpotetaan tiedon, tuotteiden ja palveluiden löytyvyyttä.</p> <p>Suosittelujärjestelmät (recommender systems) ovat sovelluksia, jotka tuottavat käyttäjälle suosituksia kohteista (tietoja, tuotteita, palveluita), joista käyttäjä todennäköisesti on kiinnostunut, ja auttavat näin käyttäjää selviytymään tarjonnan runsaudesta. Monilähteisten suosittelujärjestelmien (meta recommender systems) ideana on hyödyntää suositusten tuottamiseen useita tietolähteitä, kuten WWW-sisältöjä. WWW:n sisältöjen hyödyntämisessä on kuitenkin ongelmia, koska sisältö ei ole esitetty koneen ymmärtämässä muodossa. Semanttinen web (semantic web) on WWW:n seuraava sukupolvi, jossa sisältö esitetään koneelle ymmärrettävämmässä muodossa.</p> <p>Tässä tutkielmassa kartoitetaan miten monilähteisiä suosittelujärjestelmiä on mahdollista kehittää semanttisen webin tekniikoiden avulla siten, että useista lähteistä tulevaa tietoa voidaan hyödyntää aiempaa helpommin suositusten tuotannossa. Lisäksi esitetään mitä uusia mahdollisuuksia suosittelujärjestelmät tuovat semanttisen webin sovelluksiin. Tutkielman tärkein kontribuutio on MuseoSuomi-portaaliin toteutettu monilähteinen tietämuspohjainen suosittelujärjestelmä, joka mahdollistaa semanttisen samoilun useista eri lähteistä koostuvassa virtuaalisessa WWW-näyttelyssä.</p> <p>ACM Computing Classification System (CCS): H.5.4 [Information Interfaces and Representation]: Hypertext/Hypermedia H.3.5 [Information Storage and Retrieval]: Online Information Services I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods</p> | | | |
| Avainsanat — Nyckelord — Keywords suosittelujärjestelmät, semanttinen web, WWW | | | |
| Säilytyspaikka — Förvaringsställe — Where deposited Kumpulan tiedekirjasto, sarjanumero C- | | | |
| Muita tietoja — Övriga uppgifter — Additional information | | | |

Sisältö

| | |
|--|-----------|
| 1 Johdanto | 1 |
| 2 Suositteijärjestelmät | 3 |
| 2.1 Tarve suosittelujärjestelmille | 3 |
| 2.2 Suositteijärjestelmät | 6 |
| 2.3 Yhteistoiminnallinen suosittelu | 9 |
| 2.4 Sisältöpohjainen suosittelu | 14 |
| 2.5 Demografiapohjainen suosittelu | 16 |
| 2.6 Utiliteettipohjainen suosittelu | 17 |
| 2.7 Tietämyspohjainen suosittelu | 18 |
| 2.8 Hybridisuositteijärjestelmät | 20 |
| 3 Monilähteiset suosittelujärjestelmät | 21 |
| 3.1 MetaLens-järjestelmä | 23 |
| 3.1.1 Käyttöliittymä | 24 |
| 3.1.2 Tiedonkeruumoduulit ja tiedon fuusiointi | 25 |
| 3.1.3 Laskentamoduuli | 26 |
| 3.1.4 Hakutulosten esittäminen | 28 |
| 3.1.5 Johtopäätökset | 29 |
| 3.2 Monilähteisen suosittelun haasteet | 29 |
| 4 Semanttinen web | 32 |
| 4.1 WWW:n sovelluskäytön ongelmat | 32 |

| | | |
|----------|--|-----------|
| 4.2 | Semanttisen webin tekniikat | 34 |
| 4.2.1 | Resurssien yksilöinti: URI | 35 |
| 4.2.2 | Resurssien välinen verkko: RDF | 36 |
| 4.2.3 | RDF-skeemat ja ontologiat | 37 |
| 4.2.4 | Loogiset säännöt | 39 |
| 4.2.5 | Todistukset | 40 |
| 4.2.6 | Luottamus | 40 |
| 4.3 | Semanttisen webin erityispiirteitä | 41 |
| 4.4 | Suosittelujärjestelmät semanttisessa webissä | 43 |
| 5 | MuseoSuomen suosittelujärjestelmä | 47 |
| 5.1 | MuseoSuomi yleisesti | 47 |
| 5.2 | Tietämuspohjainen suosittelu MuseoSuomessa | 53 |
| 5.3 | MuseoSuomen suosittelujärjestelmän arviointi | 58 |
| 6 | Monilähteinen suosittelu semanttisessa webissä | 64 |
| 7 | Yhteenveto | 69 |
| | Lähteet | 71 |
| | Liitteet | |
| | 1 Liittyy tapahtumaan -suosittelemussääntö | |
| | 2 Ontodellan RDF/XML-vastaus | |

1 Johdanto

Elämme maailmassa, joka on täynnä tietoa, tuotteita ja palveluita! Kuluttajan arkipäiväinen ongelma on löytää omiin tarpeisiinsa parhaiten sopivat tuotteet ja palvelut lukuisten vaihtoehtojen joukosta, joita on saatavilla sekä sähköisten tietoliikenneverkkojen (kuten internetin ja puhelinverkkojen) että perinteisten jakelukanavien (kuten kirjastojen ja tavaratalojen) kautta. Useat tuotteet ovat lisäksi räätälöitävissä asiakkaan tarpeiden mukaan, jolloin vaihtoehtojen määrä kasvaa entisestään.

Suosittelujärjestelmien (recommender systems) ideana on auttaa käyttäjää tällaisissa arkipäivän valintatilanteissa tekemällä hänelle suosituksia kohteista (tietoja, tuotteita ja palveluita), joista hän saattaisi olla kiinnostunut. Hyvä esimerkki suosittelujärjestelmästä on Amazon.com-kauppapaikan suositukset “tuotteista, joita muutkin ihmiset ovat ostaneet.”

Perinteiset suosittelujärjestelmät hyödyntävät kuitenkin vain järjestelmän sisäistä tietoa suositusten tuottamiseen siitä huolimatta, että esimerkiksi World Wide Web (WWW) on täynnä tietoa, jolla voisi parantaa entuudestaan suosituksia. Ratkaisuna tähän on esitetty monilähteisiä suosittelujärjestelmiä (multi-source recommenders, meta recommenders), jotka hyödyntävät eri lähteissä julkaistavaa tietoa suositusten tuottamisessa, parantaen näin käyttäjän saamaa hyötyä suosittelujärjestelmästä.

Tähän mennessä esitetyissä monilähteisissä suosittelujärjestelmissä on kuitenkin ollut ongelmia WWW-sisällön hyödyntämisessä, koska WWW-sisältö on esitetty vain ihmisen ymmärtämässä muodossa, jonka tulkitseminen koneen toimesta on hankalaa. WWW:n heikko soveltuvuus sovelluskäyttöön on yleisesti tunnettu ongelma, johon ratkaisuksi on visioitu seuraavan sukupolven WWW:tä, semanttista webiä (semantic web). Semanttisen webin ideana on ratkaista WWW:n sovelluskäytön ongelmat siten, että sisältöä voidaan tulkita automaattisesti koneen toimesta. Tätä tietoa on mielekästä hyödyntää myös suosittelujärjestelmissä etenkin siten, että

suosittelevien tuotantoon voidaan käyttää useita, toisiaan täydentäviä tietolähteitä. Tämän tutkielman tavoitteena on kartoittaa, mitä uusia mahdollisuuksia semanttinen web tarjoaa monilähteisten suosittelevien järjestelmien toteuttamiseen. Erityisesti pohditaan seuraavia kysymyksiä (joita tarkennetaan luvussa 3):

1. Miten autonomisia tietolähteitä hyödynnetään suosittelevissa?
2. Miten eri lähteistä tulevat sisällöt yhdistetään?
3. Miten suositukset tuotetaan monilähteistä tietoa hyödyntäen?
4. Miten monilähteiset suositukset esitetään käyttäjälle?

Lisäksi tutkielmassa selvitetään, mitä uutta suosittelevien järjestelmät voisivat tuoda semanttisen webin sovelluksiin. Tutkielman tärkeimpänä kontribuutiona esitellään MuseoSuomi-portaaliin¹ tehty monilähteinen, semanttisen webin tekniikoita hyödyntävä tietämyspohjainen suosittelevien järjestelmä Ontodella.

Tutkielman rakenne on seuraava: Luvussa kaksi esitellään suosittelevien järjestelmien toimintaidea ja perinteisiä toteutusvaihtoehtoja. Luvussa kolme kuvataan monilähteisten suosittelevien järjestelmien toimintaidea. Luvussa neljä esitellään semanttinen web ja siihen liittyvät tekniikat. Lisäksi tuodaan esille semanttisen webin hyödyntämiseen liittyviä kysymyksiä suosittelevien järjestelmien näkökulmasta katsottuna. Luvussa viisi esitellään MuseoSuomi-järjestelmä ja sen sääntöpohjainen suosittelevien järjestelmä. Luvussa kuusi kiteytetään monilähteisen suosittelun mahdollisuudet semanttisessa webissä. Lopuksi, luvussa seitsemän on yhteenveto.

¹<http://www.museosuomi.fi>

2 Suosittelujärjestelmät

2.1 Tarve suosittelujärjestelmille

Internet ja World Wide Web (WWW) ovat mullistaneet 1990-luvulla tiedon julkaisemisen tarjoamalla välineen, jolla yksittäiset henkilöt ja organisaatiot koosta riippumatta voivat vähäisin kustannuksin julkaista tietoja ja sähköisiä palveluita koko verkottuneen maailman saataville. Julkaisemisen helppoutta kuvaa se, että WWW koostuu nykyisin yli 11,5 miljardista sivusta [GS05]. Digitalisoituminen koskee myös esimerkiksi mobiilipalveluita sekä televisiota, jotka molemmat tukevat sekä WWW-tyyppisiä palveluita (WAP [WAP] mobiililaitteilla ja superteksti-tv digi-tv:ssä) että sovelluksia (esimerkiksi paikannukseen perustuvat palvelut mobiililaitteilla ja digi-tv:ssä elektroninen ohjelmaopas (EPG) sekä tulevaisuudessa mm. interaktiiviset tv-ohjelmat ja tilausvideopalvelut).

Digitalisoituminen vähentää tuotantokustannuksia, koska yhä useampi laite perustuu samoihin perustekniikoihin, kuten esimerkiksi ohjelmoitavaan tietokoneeseen, Internet-protokollaan, HTML- [HTM], WAP- [WAP] ja muihin XML-perustaisiin sisällönkuvauskieliin [XML], Java-sovelluksiin [Jav], MPEG-videoon [Mpe] jne. Tämän seurauksena sisällöntuotanto helpottuu ja sisältöjen siirto jakelukanavien välillä on helpompaa kuin analogisella aikakaudella. Sisältöä on yhä enemmän saatavilla kaikkien jakelukanavien kautta.

Kuluttajan arkipäiväinen ongelma on löytää omiin tarpeisiinsa parhaiten sopivat kohteet (tiedot, tuotteet ja palvelut) lukuisten vaihtoehtojen joukosta, joita on tarjolla sekä sähköisessä muodossa että perinteisten jakelukanavien, kuten kirjastojen ja tavaratalojen, kautta. Useat tuotteet ovat lisäksi räätälöitävissä asiakkaan tarpeiden mukaan, mikä toisaalta helpottaa tuotteen valintaa (tuote voidaan sovittaa käyttäjän mukaan eikä käyttäjä tuotteeseen), mutta mikä toisaalta entisestään lisää

tarjolla olevien vaihtoehtojen määrää.

Massaräätälöinti (mass-customization) [II93] on tapa tuottaa yksilöllisiä tuotteita massatuotantona siten, että saavutetaan sekä massatuotannosta saatavat kustannushyödyt että räätälöinnistä saatava lisäarvo. Tämä lisäarvo kasvattaa optimitilanteessa sekä tuotteen kysyntää että mahdollistaa korkeamman hinnoittelun verrattuna kilpaileviin tuotteisiin. Massaräätälöinnin mahdollistaa kehittynyt informaationhallinta (knowledge management) ja tilauspohjainen tuotanto, jossa asiakkaan yksilölliset tarpeet välitetään tuotantoketjun alkupäähän siten, että tuloksena syntyy yksilöllinen tuote.

Esimerkiksi autotehtaissa tuotetaan tyypillisesti kullekin ostajalle räätälöity auto, johon asiakas on autoa tilatessaan valinnut esimerkiksi haluamansa värin, moottorin ja lisävarusteet. Informaatiopainotteisissa tuotteissa, kuten WWW-sivuissa, elokuvissa ja kirjoissa massaräätälöintiä voidaan tehdä sekä sisällön että jakelukanavien osalta. Esimerkiksi WWW-palvelun sisältö voidaan räätälöidä kullekin käyttäjälle erikseen, jotta palvelu vastaisi mahdollisimman hyvin käyttäjän yksilöllisiä tarpeita. Elokuvia voidaan puolestaan levittää useiden eri jakelukanavien kautta, kuten esimerkiksi elokuvateattereissa, vuokravideon, tv-kanavilla ja tilausvideopalvelussa. Elokuvan sisältö ei tällöin yleensä muutu jakelukanavien myötä, mutta eri jakelukanavien avulla katsoja voi räätälöidä itselleen ajan ja paikan osalta itselleen parhaiten sopivan tavan katsoa tietty elokuva.

Kuluttaja ei useinkaan pysty käytettävissä olevien resurssien puitteissa perehtymään kovin syvällisesti tarjolla oleviin vaihtoehtoihin ennen valintapäätöksen tekemistä. Lisäksi kuluttajan on usein käytännössä hankalaa saada itselleen valintojen tekemisen kannalta olennaisia tietoja, koska esimerkiksi tuotteen myyjälle ei välttämättä ole edullista kertoa puolueettomasti tuotteen hyvistä ja huonoista puolista. Puolueettomampaa tietoa kuten lehtiartikkeluita tai ystävien kommentteja ei puolestaan välttämättä ole sillä hetkellä käytössä, kun valintapäätöksen tekeminen on ajankoh-

taista, esimerkiksi elokuvavuokraamossa käydessä.

WWW:n osalta ratkaisu tiedonhakuun ovat hakukoneet, kuten Google [Goo], ihmisvoimin ylläpidetyt verkkohakemistot, kuten Open directory project dmoz.org [Dmo] ja Yahoo [Yah], sekä aihekohtaiset portaalit, kuten Suomi.fi [Suo]. Yksittäiset WWW-palvelut voivat myös olla siinä määrin kattavia sisällöltään, että käyttäjä saa kaiken haluamansa yhdestä palvelusta. Esimerkiksi Amazon.com-verkkokauppa [Ama] myy mm. kirjoja, elokuvia, tietokoneohjelmia, kodinkoneita ja vaatteita. Kaikissa edellä kuvatuissa tapauksissa oleellista on, että käyttäjälle tarjotaan tehokkaita keinoja löytää palvelun tarjonnasta hänelle mielenkiintoinen sisältö.

Hakukoneiden haasteena on tarjota hakutuloksina käyttäjän hakua vastaavia sivuja (tai pikemminkin sivustoja [Hea00]), jotka ovat sekä tietosisällöltään olennaisia että laadultaan hyviä. Googlen tapauksessa tämä on ratkaistu siten, että sivujen indeksoinnissa hyödynnetään, paitsi yksittäisen sivun tekstisisältöä, myös tietoa siitä, miten yksittäinen sivu on linkitetty muihin sivuihin. Sivujen välisten linkkisuhteiden katsotaan heijastavan implisiittisesti WWW-käyttäjien yleistä mielipidettä siitä, mikä on hyvää, linkittämisen arvoista sisältöä [PBMW98].

Hakukoneiden vahvuus on se, että ne kattavat huomattavan osan WWW:n sisällöstä, koska indeksointi tapahtuu automaattisesti. Käytännössä kaikki sivut, joihin johtaa linkkejä muualta, ovat haettavissa. Koska sivut indeksoidaan merkkijonojen avulla ilman ymmärrystä sivun sisällöstä, on hakutuloksessa kuitenkin paljon myös epäolennaisia sivuja, joissa syystä tai toisesta esiintyy hakuun käytetty merkkijono.

Ihmisvoimin ylläpidetyt verkkohakemistot ovat laadultaan tasalaatuisempia, koska sivujen sisältöä tulkitaan ihmisen toimesta ennen niiden lisäämistä hakemistoon. Suuresta työmäärästä johtuen verkkohakemistot ovat kuitenkin väistämättä suppeampia kuin automaattiseen indeksointiin perustuvat hakukoneet. Verkkohakemiston toimitus ottaa myös sivuja valitessaan väistämättä sisällöllistä kantaa siihen,

mikä on olennaista sisältöä WWW:ssä ja mikä ei. Automaattiseen indeksointiin perustuvat hakukoneet indeksoivat oletusarvoisesti kaiken sisällön. (Toki myös hakukoneiden tuloksia voidaan ihmisen toimesta peukaloida, kuten esimerkiksi Google on hiljattain tehnyt Kiinassa näkyvän Googlen osalta, jossa poliittisesti epäkorrekti sisältö sensuroidaan.)

Aihekohtaiset portaalit ovat tyypillinen tapa jäsentää WWW:n sisältöä. Ne ovat muodoltaan usein samantyyppisiä kuin edellä kuvatut hakemistot, mutta keskittyvät rajattuun aihealueeseen. Portaalien ideana on toimia aloituspisteenä Internet-käytölle ja tiedonhauille. Niiden sisältö koostuu tyypillisesti linkkikokoelmista sekä toimitetusta sisällöstä [Sid05]. Portaalien ennakoidaan olevan käytössä myös uusissa päätelaitteissa kuten digi-tv:ssä [Sor04].

Hyvä esimerkki yksittäisen verkkopalvelun tarjoamista tehokkaista navigaatio- ja hakutoiminnoista on Amazon.com-verkkokauppa [Ama], jonka tuotevalikoima koostuu miljoonista tuotteista. Amazon.com:in tavoitteena on tarjota kullekin asiakkaalle (virtuaalista) henkilökohtaista palvelua samaan tapaan kuin pikkukaupungin kauppias tuntee jokaisen käyttäjän tarpeet yksilöllisesti ja osaa suositella hänelle sopivia tuotteita [Whe00]. Tuotteiden löytämisen helpottamiseksi Amazon.com:ssa on käytetty paitsi tuotteiden luokittelua taksonomioihin myös tuotteiden automaattista suositelua esimerkiksi kävijöiden ostosten perusteella (“tämän tuotteen ostaneet ovat pitäneet myös seuraavista tuotteista...”).

2.2 Suosittelevat järjestelmät

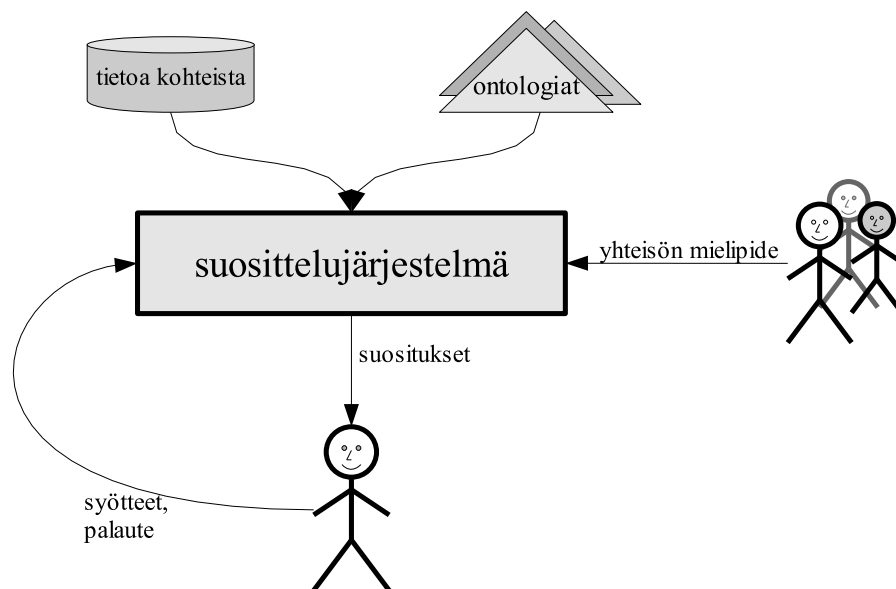
Tiedonhakujärjestelmissä (information retrieval systems) [BYRN99] tavoitteena on palauttaa käyttäjälle ne dokumentit, jotka vastaavat käyttäjän tarvetta parhaiten. Perinteisissä tiedonhakujärjestelmissä dokumentin osuvuutta (relevanssi) arvioidaan pääsääntöisesti vertaamalla käyttäjän antamia hakutermejä dokumentin sisältöön

siten, että mitä merkityksellisemmässä asemassa käyttäjän antamat hakutermit ovat dokumentissa, sitä osuvampi dokumentti on kyseessä. Tiedonhakujärjestelmistä on olemassa lukuisia erilaisia variaatioita, jotka soveltuvat esimerkiksi erilaisten sisältöjen hakuun (teksti-, kuva-, äänidokumentit) ja joissa käytetään erilaisia tapoja järjestää hakutulokset relevanssijärjestykseen. Tyypillinen piirre tiedonhakujärjestelmissä on se, että ne pohjautuvat pääsääntöisesti sisällön analysointiin.

Suosittelujärjestelmät (recommender systems) laajentavat perinteisiä tiedonhakujärjestelmiä siten, että pääpainona on tuottaa käyttäjälle *käyttäjäkohtaisia, kiinnostavia ja hyödyllisiä* hakutuloksia tai ehdotuksia perustuen järjestelmän tietoihin käyttäjästä ja hänen tarpeistaan [RIS⁺94, RV97, Bur02, Sch01]. Suosittelujärjestelmät ovat erityisen hyödyllisiä silloin, kun tarjolla olevien vaihtoehtojen määrä on niin suuri, ettei käyttäjällä ole mahdollisuuksia omakohtaisesti käydä läpi koko aineistoa, kuten esimerkiksi Amazon.com:in miljoonien tuotteiden tuotevalikoimaa tai Internetin tietosisältöä. Mitä paremmin käyttäjälle pystytään tarjoamaan juuri häntä kiinnostavat kohteet, sitä paremmin käyttäjä myös saavuttaa omat tavoitteensa, kuten esimerkiksi onnistuneen elokuvaillan. Mitä paremmin palvelun tarjoaja, esimerkiksi kauppapaikan pitäjä, pystyy palvelemaan asiakkaitaan, sitä todennäköisemmin nämä käyttävät palvelua uudelleen.

Suosittelujärjestelmien toiminta perustuu tietoon käyttäjistä ja suositeltavista kohteista, joiden perusteella tuotetaan aktiiviselle käyttäjälle suosituksia. Kuva 1 esittää suosittelujärjestelmän syötteet ja tulosteet yleisellä tasolla. Eri suosittelujärjestelmissä hyödynnetään kuvan esittämistä tiedoista yleensä vain osaa.

Tiedot suositeltavista kohteista voivat olla esimerkiksi otsikko, sisältökuvaus ja luokittelutietoa, kuten elokuvan genre. Lisäksi saatetaan hyödyntää yleisempää tietoa aihealueesta (ontologiat), jolla kuvataan esimerkiksi elokuvien osalta, miten elokuvat jakaantuvat genre-luokittelun mukaan ryhmiin. Suositusten tuotantoon voidaan hyödyntää myös muiden käyttäjien mielipiteitä, jotka voivat olla esimerkiksi numero-



Kuva 1: Suositelujärjestelmän syötteet ja tulosteet, Schaferia et al. mukailen [SKR01].

tai tekstiarvioita suositeltavista kohteista sekä käyttäjien ostohistoriatietoja (mitä kohteita henkilö on ostanut).

Aktiiviselta käyttäjältä, jolle suosituksia tehdään, voidaan saada syötteenä esimerkiksi navigaatiotietoa, arvioita ja ostohistoriatietoa [SKR01]. Navigaatiotieto voi olla joko eksplisiittistä tai implisiittistä. Eksplisiittinen syöte syntyy silloin, kun käyttäjän ensisijainen tarkoitus on vaikuttaa suosittelujärjestelmään, esimerkiksi arvioimalla jonkin kohteen [Nic97]. Implisiittinen tieto puolestaan syntyy käyttäjän muun toiminnan ohessa ilman, että käyttäjän tavoitteena olisi vaikuttaa suosittelujärjestelmään. Esimerkiksi kirjan ostaminen tai tiedoston tallentaminen voidaan tulkitaa vahvoiksi implisiittisiksi merkeiksi siitä, että käyttäjä on kiinnostunut kyseisistä kohteista.

Useissa suosittelujärjestelmissä käyttäjälle muodostetaan käyttäjäprofiili, johon tallennetaan suositusten tuottamisen kannalta tarpeellisia tietoja. Profiili voi sisältää esimerkiksi käyttäjän antamia numeroarvioita suositeltavista kohteista (taulukko 1).

| käyttäjä | kirja 1 | kirja 2 | kirja 3 | kirja 4 | ... |
|----------|---------|---------|---------|---------|-----|
| Kalle | - | 1.0 | 0.5 | 0.0 | ... |

Taulukko 1: Esimerkki käyttäjäprofiilista, jossa käyttäjän mielipide kustakin kirjasta ilmaistaan numeroiden avulla. Kirja, jota käyttäjä ei ole arvioinut, on merkitty viivalla.

Profiili voi numeroarvioiden sijaan koostua myös merkkijonoista tai muista tietotyypeistä, jotka kuvaavat käyttäjän kiinnostusta. Profilit ovat tyypillisesti pitkäikäisiä eli niitä täydennetään ja hyödynnetään suosittelussa käyttökerrasta toiseen. Esimerkiksi Amazon.com tallentaa tiedot käyttäjän kaikista ostoksista, jolloin tätä tietoa voidaan hyödyntää myös suosittelussa.

Käyttäjakohtaisiin, kiinnostaviin ja hyödyllisiin suosituksiin on pyritty useilla eri keinoilla, joista osa on hyvin lähellä tyypillisiä sisältöpohjaisia sisällönhakujärjestelmiä, kun taas toiset perustuvat täysin käyttäjältä saatavaan palautteeseen. Alunperin suosittelujärjestelmillä tarkoitettiin nimenomaan yhteistoiminnalliseen suositteluun perustuvia järjestelmiä [RV97], mutta myöhemmin suosittelujärjestelmien käsitteeseen on luettu myös muita suosittelumenetelmiä, kuten esimerkiksi sisältöpohjainen ja tietämuspohjainen suosittelu [Bur02].

2.3 Yhteistoiminnallinen suosittelu

Yhteistoiminnallinen suosittelu (collaborative filtering) perustuu siihen, että järjestelmä pyrkii tunnistamaan samanhenkiset käyttäjät ja suosittelemaan heille kohteita, joista muut samoista aiheista kiinnostuneet käyttäjät ovat pitäneet [RIS⁺94, Bur02]. Tähän ideaan perustuu esimerkiksi Amazon.com:in suositukset kirjoista, joista toiset käyttäjät ovat pitäneet (ostaneet). Yhteistoiminnallinen suosittelu on eri suosittelumenetelmistä tunnetuin. Siihen perustuu mm. uutisryhmien suositte-

| käyttäjä | kirja 1 | kirja 2 | kirja 3 | kirja 4 | ... |
|----------|---------|---------|---------|---------|-----|
| Kalle | - | 1.0 | 0.5 | 0.0 | ... |
| Matti | 1.0 | 0.3 | 0.5 | - | ... |
| Liisa | 1.0 | - | 0.5 | 1.0 | ... |
| Maija | - | - | - | 1.0 | ... |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

Taulukko 2: Esimerkki järjestelmään kerätystä profilitiedosta, josta ilmenee, minkä arvosanan kukin käyttäjä on antanut kullekin kirjalle. Kirjat, joita käyttäjä ei ole arvioinut, on merkitty viivalla.

luun tarkoitettu GroupLens-järjestelmä, joka on ensimmäisiä toteutettuja yhteistoiminnallisia suosittelujärjestelmiä [RIS⁺94].

Yhteistoiminnallisessa suosittelussa käyttäjät, joiden profilit muistuttavat toisiaan muodostavat mielipidenaapurustoja (neighbourhood), joissa naapuruston jäsenten välillä voidaan tehdä suosituksia. Mikäli yksi naapuruston jäsenistä ei ole arvioinut tiettyä kohdetta, voidaan muiden naapureiden antamien arvioiden perusteella päätellä, mitä mieltä kohteena oleva henkilö olisi kyseisestä kohteesta.

Yhteistoiminnallisessa suosittelussa käyttäjäprofiili esitetään tyypillisesti vektorina, joissa kullekin käyttäjän arvioimalle kohteelle on annettu kohteen kiinnostavuutta kuvaava luku (esimerkiksi reaalityyppi välillä $[0, 1]$). Taulukko 2 esittää esimerkkiä profilivektoreista, jossa kukin käyttäjä on eksplisiittisesti tai implisiittisesti antanut numeroarvion kirjoista. Kirjat, joita yksittäinen käyttäjä ei ole arvioinut, ovat kirjoja, joita hänelle mahdollisesti jatkossa suositellaan.

Naapuruston luominen voidaan tehdä useilla eri menetelmillä. Ensimmäiset yhteistoiminnalliset suosittelujärjestelmät perustuivat siihen, että käyttäjät itse kertoivat, kenen suosituksiin luottavat [GNOT92]. Tällainen menetelmä saattaa toimia

hyvin esimerkiksi työyhteisön sisällä, mutta laajemmassa Internet-palvelussa, kuten Amazon.com:ssa, käyttäjät tuskin tuntevat toisiaan, jolloin käyttäjien on vaikea nimetä samanhenkisiä muita käyttäjiä. Automaattinen yhteistoiminnallinen suosittelu perustuu siihen, että suosittelujärjestelmä tunnistaa ohjelmallisesti samanhenkiset käyttäjät, joiden välillä suosituksia tehdään [RIS⁺94].

Jotta kahta käyttäjäprofiilia voisi verrata toisiinsa, tulee niiden sisältää yhteisiä kohteita. Tulee siis olla voimassa tilanne $|P_a \cap P_b| > 0$, jossa P_a ja P_b ovat käyttäjien a ja b arvioimien kohteiden muodostamat joukot. Mitä enemmän yhteisiä kohteita profileilla on, sitä varmemmin suosituksia voidaan tuottaa. Tämän vuoksi suosittelujärjestelmissä on yleensä määrätty alaraja sille, kuinka monta yhteistä kohdetta profileilla tulee olla, jotta niitä ylipäätään vertailtaisiin toisiinsa.

Kahden profiilin vertailu tapahtuu laskemalla profiilien välinen samankaltaisuus (etäisyys, korrelaatio tai similariteetti) jotain sopivaa laskentamenetelmää käyttäen. Kaksi yleisesti käytettyä menetelmää profiilien vertailuun ovat Pearsonin korrelaatiokaava (kaava 1) sekä kosinilauseke (kaava 2) [BHK98]. Kaavojen tuloksena saadaan lukuarvo, joka ilmaisee kuinka hyvin vertailtavat kohteet vastaavat toisiaan skaalalla nollasta (eivät vastaa) yhteen (vastaavat täysin toisiaan). Korrelaation yleinen tulkinta on kuvattuna taulukossa 3.

$$r(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

Kaavassa 1 x ja y tarkoittavat kahta profiilivektoria. Merkinnät \bar{x} ja \bar{y} tarkoittavat x :n ja y :n alkoiden keskiarvoa.

Pearsonin kaavan etuja on se, että sillä voidaan laskea myös profiilien välinen negatiivinen korrelaatio toisin kuin kosinilausekkeen avulla. Negatiivinenkin korrelaatio voi toimia suosittelun pohjana. Jos kahden käyttäjän välillä on voimakas negatiivi-

| korrelaatiokerroin | korrelaation tulkinta |
|----------------------|------------------------|
| $ x = 1, 0$ | lineaarinen riippuvuus |
| $ x = [0, 8, 1, 0[$ | voimakas |
| $ x = [0, 6, 0, 8[$ | huomattava |
| $ x = [0, 3, 0, 6[$ | kohtalainen |
| $ x = [0, 0, 3[$ | merkityksetön |

Taulukko 3: Korrelaatiokertoimen yleinen tulkinta [STK⁺92]. Suosittelevjärjestelmässä on joissain tapauksissa pieniä toteutuskohtaisia poikkeamia rajojen suhteen.

nen korrelaatio, voidaan tämän perusteella ennustaa, että kaikki mistä toinen ei ole pitänyt, saattaa kiinnostaa toista osapuolta. Pearsonin korrelaatiokaavaa pidetään yhteistoiminnallisen suosittelun profiilivälisen vertailun de-facto-laskentamenetelmänä [ZG05].

Kosinilauseen (kaava 2) avulla mitataan kahden vektorin välistä kulmaa, jossa vektoreina esitetään vertailtavat käyttäjäprofiilit. Vektorit vastaavat toisiaan täydellisesti, mikäli $\cos(\vec{x}, \vec{y}) = 1$ ja eivät vastaa laisinkaan toisiaan, jos $\cos(\vec{x}, \vec{y}) = 0$. Kosinilausekkeen käyttäminen suosittelujärjestelmässä on todennäköisesti lähtöisin tiedonhakuprosessien, joissa dokumentteja verrataan tyypillisesti toisiinsa dokumenteista muodostettujen sanafrekvenssivektoreiden avulla.

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|} = \frac{\sum_{i=1}^n (x_i y_i)}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (2)$$

Pearsonin korrelaatiokaava (kaava 1) on muunnettavissa kosinilauseen (kaava 2) muotoon siten, että keskiarvot \bar{x} ja \bar{y} korvataan vakiolla z , jonka arvoksi annetaan neutraali pistemäärä. Esimerkiksi skaalalla 1-5 neutraali pistemäärä on keskiarvo 3. Tämän jälkeen z -vähennys suoritetaan alkuperäisiin vektoreihin [SN00].

Profiilien vertailu voidaan tehdä joko reaaliaikaisesti tai eräajona. Reaaliaikaisen

suosittelemisen ongelmana voi olla hitaus, mikäli profiileja ja kohteita on paljon. Tästä johtuen tärkeä tavoite suosittelujärjestelmien tutkimuksessa on ollut kehittää vakioajassa toimivia suosittelujärjestelmiä, joissa suositusten laatu ei kuitenkaan heikkenisi [SKR01, GRGP01].

Yhteistoiminnallisen suosittelun suurin etu on siinä, että arvioinnin kohteena voi olla mitä tahansa sisältöjä, koska tietokoneen ei tarvitse käsitellä kohteiden sisältöä tuottaakseen suosituksia (toisin kuin esimerkiksi sisältöpohjaisessa suosittelussa) [Bur02, Her00]. Kohteet voivat olla esimerkiksi tekstidokumentteja, videoita tai abstraktimpia asioita (kuten vitseissä [GRGP01]).

Toinen etu on se, että yhteistoiminnallisessa suosittelussa suosittelu voi perustua monimutkaisiin ja vaikeasti ilmaistaviin mielipiteisiin, kuten makukysymyksiin ja laatuun [Her00]. Yhteistoiminnallisessa suosittelussa käyttäjän ei tarvitse pystyä määrittelemään kohteiden sisältöön liittyviä kriteereitä (esimerkiksi mitkä ominaisuudet kuvaavat ”hyvän” elokuvan), vaan suosittelu perustuu samanhenkisten muiden käyttäjien mielipiteisiin. Tästä mielipideperustaisuudesta seuraa toisaalta sellainen ongelma, että käyttäjälle on hankalampi tuottaa perusteluita tehdyille suosituksille [HKR00]. Perusteluiden tuottaminen on sitä tärkeämpää mitä suurempia taloudellisia tai muita riskejä päätökseen liittyy.

Yhteistoiminnallisen suosittelun etuja on myös mahdollisuus löytää ennalta arvaamattomia suosituksia kohteista, jotka eivät esimerkiksi sisällöllisesti muistuta toisiaan, mutta joista samanhenkiset käyttäjät ovat pitäneet [Her00]. Jos esimerkiksi käyttäjä ei tyypillisesti pidä sci-fi-elokuvista, sisältöpohjainen suosittelujärjestelmä rajaa kaikki sci-fi-elokuvat suositusten ulkopuolelle. Yhteistoiminnallinen suosittelujärjestelmä puolestaan perustuu samanhenkisten käyttäjien mielipiteisiin, joten jos jokin tietty sci-fi-elokuva on poikkeuksellisesti useimpien samanhenkisten sci-fiä vieroksuviiden käyttäjien mielestä hyvä, tätä kyseistä elokuvaa voidaan suositella käyttäjälle, joka muuten ei pidä sci-fi:stä.

Yhteistoiminnallisen suosittelun ongelmia ovat puolestaan uuden käyttäjän ja uuden kohteen ongelmat [Bur02]. Uuden käyttäjän ongelma tarkoittaa tilannetta, jossa uusi käyttäjä ei vielä ole arvostellut yhtään tai kovin montaa kohdetta. Tästä seuraa se, että profilien vertailuun perustuva suosittelu toimii huonommin kuin normaalkokoisella profiililla. Tämä on haasteellista mm. sen vuoksi, että suosittelujärjestelmä tuottaa heikkotasoisimpia suosituksia silloin, kun käyttäjä on vasta aloittamassa järjestelmän käyttöä ja todennäköisesti hylkää sen helpoiten. Uuden käyttäjän ongelmaan liittyy myös tilanne, jossa suosittelujärjestelmä on vastikään julkaistu, eikä käyttäjien profileja ole vielä tarpeeksi.

Uuden kohteen ongelma tarkoittaa tilannetta, jossa järjestelmään on lisätty uusi kohde, mutta kukaan ei ole vielä arvioinut sitä [Bur02]. Tästä seuraa, ettei kyseistä kohdetta myöskään suositella kenellekään. Järjestelmän toiminnan kannalta oleellista on keksiä keinot, joilla uudetkin kohteet saadaan mukaan arvosteluun. Tämä voidaan ratkaista esimerkiksi palkitsemalla käyttäjää uusien kohteiden arvioimisesta tai täydentämällä yhteistoiminnallista suosittelua muilla suosittelumenetelmillä, kuten myöhemmin ilmenee.

Koska yhteistoiminnallinen suosittelu perustuu siihen, että järjestelmän käytössä on lukuisten käyttäjien profileja, joiden pohjalta suosituksia voidaan tuottaa muille käyttäjille, on järjestelmän menestymisen kannalta tärkeää, että suosittelujärjestelmä houkuttelee tarpeeksi käyttäjiä. Muuten jokaiselle käyttäjälle ei välttämättä löydy samanhenkisiä muita käyttäjiä.

2.4 Sisältöpohjainen suosittelu

Sisältöpohjaisen suosittelun (content-based recommendation) ideana on pyrkiä löytämään dokumenttien sisältöä analysoimalla luokittelusääntöjä, joiden avulla voitaisiin arvioida käyttäjälle entuudesta tuntemattomien kohteiden kiinnostavuut-

ta [SKR01, Bur02]. Esimerkiksi tekstidokumenttien osalta voidaan etsiä sanoja ja fraaseja, jotka esiintyvät tyypillisesti käyttäjäprofiilin perusteella hyviksi arvioiduissa dokumenteissa. Vastaavasti huonoksi arvioiduista dokumenteista voidaan tunnistaa niitä tyypillisesti kuvaavia piirteitä. Näiden tietojen perusteella voidaan tämän jälkeen tehdä ennusteita siitä, mitä mieltä käyttäjä on entuudestaan tuntemattomista uusista kohteista. Tekstiä sisältämättömien kohteiden kuten valokuvien ja videoiden osalta sisältöpohjainen suosittelu voi perustua paitsi kohteiden analysointiin (esimerkiksi valokuvan väri, muoto jne) myös kohteen metatietoihin kuten vapaa tekstikuvaus kohteesta.

Esimerkiksi Fab-järjestelmä [BS97] sisältää WWW-sivujen suositteluun tarkoitetun sisältöpohjaisen suosittelujärjestelmän. Järjestelmä etsii WWW-sivuja, jotka vastaavat sisällöltään käyttäjän sisältöprofiilia. Jos sopivia sivuja löytyy, niitä suositellaan käyttäjälle. Fabin sisältöprofiili kuvataan painotettuina merkkijonoina, joissa paino kuvaa sitä, kuinka merkityksellinen kyseinen sana on profiilissa. Profiili voi sisältää esimerkiksi merkkijonon "tax-payer" painolla 0,23. Käyttäjä voi antaa palautetta sisältöpohjaisesti suositelluista WWW-sivuista 7-portaisella skaalalla. Palautteen perusteella muutetaan sisältöprofiilin sanoja ja painokertoimia niiltä osin, kuin sanat esiintyvät arvioidulla WWW-sivulla.

Fab on toteutettu agenttipohjaisesti siten, että kullakin käyttäjällä on oma agenttinsa, joka etsii aktiivisesti käyttäjää kiinnostavia kohteita hyödyntäen yhden tai useamman kerääjäagentin palveluita, jotka suorittavat varsinaisen WWW-sivujen etsinnän Internetissä. Käyttäjältä saatavan palautteen perusteella agentit muuttavat toimintaansa. Jos tietyt kerääjäagenttin keräämät sisällöt saavat säännönmukaisesti huonoa palautetta käyttäjiltä, kyseinen agentti sammutetaan. Vastaavasti, jos jokin kerääjäagentti menestyy hyvin, voidaan se monistaa useaksi uudeksi agentiksi. Sisältöpohjaisen suosittelun erityinen vahvuus verrattuna yhteistoiminnalliseen suositteluun on se, että myös sellaisia kohteita voidaan suositella, joita kukaan muu

käyttäjä ei ole vielä nähnyt. Sisältöpohjaisen suosittelun heikkous on se, että käyttäjälle saatetaan suositella sellaisia dokumentteja, jotka muistuttavat sisällöltään liiankin paljon aiemmin suositeltuja dokumentteja. Käyttäjälle ei myöskään pystytä perustapauksessa tuottamaan suosituksia dokumenteista, jotka olisivat käyttäjälle kiinnostavia, mutta joiden sanat eivät täsmää käyttäjän sisältöprofiiliin. Näin käy esimerkiksi silloin, jos dokumentin kieli on eri kuin sisältöprofiilissa.

Sisältöpohjainen suosittelu pohjautuu pitkälti tiedonhakumenetelmiin ja tällaisia suosittelujärjestelmiä on toteutettu useilla eri tekniikoilla, kuten päätöspuilla, neuroverkoilla ja vektoripohjaisilla menetelmillä [BHK98].

2.5 Demografiapohjainen suosittelu

Demografiapohjainen suosittelu (demographic recommender systems) perustuu siihen, että käyttäjä kuvataan demografisten ominaisuuksien, kuten esimerkiksi iän, sukupuolen tai varallisuuden avulla [Bur02]. Suosituksia voidaan tämän jälkeen tuottaa siten, että tiettyä tuotetta suositellaan kaikille niille henkilöille, joilla on valitut demografiset ominaisuudet, kuten esimerkiksi alle 30-vuotiaille naisille, joiden vuositulot ylittävät 30 000 euroa.

Kruwlich on kehittänyt demografiapohjaisen WWW-sivujen suosittelujärjestelmän, joka perustui tilastotietoon yli 40 tuhannesta Yhdysvaltojen kansalaisesta [Kru97]. Tilasto sisälsi kunkin henkilön osalta elämäntapaan ja kulutukseen liittyviä tietoja, kuten esimerkiksi pelaako henkilö golfia ja katsooko hän HBO-televisiokanavaa. Yhteensä näitä ominaisuuksia oli yli 600. Tietojen perusteella Yhdysvaltojen kansalaiset voitiin jakaa 62 demografiseen ryhmään.

Kruwlichin suosittelujärjestelmässä käyttäjältä kysytään muutamia profiloivia kysymyksiä, joiden avulla selvitetään, mihin demografiseen ryhmään käyttäjä kuuluu. Kysymykset ja niiden järjestys valitaan siten, että oikea ryhmä löytyisi mahdolli-

simman nopeasti. Esimerkiksi mikäli henkilö vastaa, että hän pelaa tennistä, katsoo HBO-kanavaa, lukee lehdestä talousuutiset ja asuu pienkaupungissa, voidaan aineiston perusteella päätellä, että kyse on todennäköisesti menestyshakuisesta “valkokaulustyöntekijästä”. Kyseiset henkilöt ovat tyypillisesti kiinnostuneita talouteen liittyvistä palveluista, kuuntelevat progressiivista rokkia ja lukevat mielellään tiede- ja tekniikkalehtiä. Järjestelmä suosittelee tätä tietoa hyödyntäen käyttäjälle WWW-sivuja, joista hän saattaisi olla kiinnostunut.

Demografisen suosittelu mahdollistaa käyttäjätietojen yleistämisen siten, että käyttäjät kuvataan esimerkiksi edellä kuvatun 62 eri demografiaryhmän avulla eikä esimerkiksi yksittäisten kohteiden avulla, kuten yhteistoiminnallisessa suosittelussa perinteisesti tapahtuu. Kun mahdollisia vaihtoehtoja on vähemmän, on käyttäjäprofiilien vertaileminen helpompaa. Demografiapohjaista suosittelua voidaan hyödyntää myös esimerkiksi WWW-markkinoinnissa siten, että käyttäjälle näytetään sellaisia mainoksia, jotka todennäköisimmin kiinnostavat häntä [Kru97].

Demografisen suosittelun haaste on se, että stereotyyppiset käyttäjäryhmät eivät aina ennakoita sitä, mistä kohteista käyttäjä on kiinnostunut ja mistä ei. Esimerkiksi köyhä iäkäs nainen voi olla kiinnostunut nopeista urheiluautoista ja nuori rikas mies puolestaan puutarhanhoidosta.

2.6 Utiliteettipohjainen suosittelu

Utiliteettipohjainen suosittelu (utility-based recommendation) perustuu siihen, että kullekin kohteelle lasketaan hyödyllisyysarvo (utiliteetti), joka kuvaa kuinka hyödyllinen kyseinen kohde on käyttäjälle [Bur02]. Esimerkiksi verkkokaupan käyttäjä saattaa preferoida nopeaa toimitusta enemmän kuin tuotteen hintaa. Mikäli toinen tuote löytyy ostohetkellä varastosta ja toista ei, voidaan utiliteettifunktiossa preferoida nopeaa toimitusaikaa. Käyttäjäkohtainen utiliteettifunktio voidaan määritellä

lä esimerkiksi interaktiivisella käyttöliittymällä, jonka avulla käyttäjä voi valita ja painottaa haluamiaan ominaisuuksia riippuen tarpeistaan [Gut98].

Interaktiivisen utiliteettipohjaisen suosittelun hyviä puolia on se, että järjestelmä mukautuu välittömästi käyttäjän muuttuviin priorisointeihin tuotteiden ominaisuuksiin liittyen. Lisäksi utiliteettifunktiossa voidaan huomioida tuotteisiin epäsuorasti liittyviä tietoja kuten esimerkiksi varastotilannetta tai toimitusaikoja. Interaktiivinen utiliteettipohjainen suosittelu ei myöskään sisällä oppimisvaihetta, koska käyttäjältä pyydetään eksplisiittiset priorisoinnit etsittäville kohteille. Heikkoutena on se, että käyttäjältä vaaditaan syötteitä utiliteettifunktion luomiseksi toisin kuin implisiittiseen tietoon perustuvissa menetelmissä.

2.7 Tietämuspohjainen suosittelu

Tietämuspohjaisessa suosittelussa (knowledge-based recommendation) ideana on mallintaa funktionaalista tietoa siitä, miten tarjolla olevat kohteet vastaavat käyttäjien tarpeisiin [Bur02, Bur00]. Esimerkiksi ravintolasuosittelujärjestelmässä ravintolat voidaan kuvata ominaisuuksien, kuten tyylin ja sijainnin, avulla. Näitä ominaisuuksia hyödyntäen voidaan funktionaalisenä tietona määritellä, mitkä loogiset ehdot ravintolan tulee täyttää, jotta se olisi ”hiljainen ravintola, jossa on merinäköala”.

Esimerkiksi Entree-nimisessä tietämuspohjaisessa ravintolasuosittelujärjestelmässä käyttäjä voi etsiä ravintoloita ominaisuuksien perusteella joko esimerkkilähtöisesti tai nimeämällä halutut ominaisuudet [Bur00]. Esimerkkilähtöisessä haussa käyttäjä nimeää haun lähtökohdaksi tietyn ravintolan, jolloin järjestelmä palauttaa aihealueen tietämyksensä perusteella ominaisuuksiltaan lähellä olevia muita ravintoloita. Tämän jälkeen käyttäjä voi ohjata järjestelmää haluamaansa suuntaan antamalla jatkoehdoiksi funktionaalisia sääntöjä, kuten esimerkiksi ”samanlainen, mutta halvempi” tai ”samanlainen, mutta kokeellisempaa ruokaa”. Entree mahdollistaa myös

ravintoloiden hakemista ominaisuuksien avulla siten, että käyttäjä määrittelee esimerkiksi haluamansa kulinartistisen tyyli suunnan, hintatason ja tunnelman. Tämän haun tuloksena tulevia kohteita hän voi tarkentaa edellä kuvatulla esimerkkilähtöisellä haulla, kunnes löytää sopivan kohteen.

MuseoSuomi-järjestelmän (luku 5) tietämuspohjaisessa suosittelujärjestelmässä puolestaan määritellään loogisten sääntöjen ja aihealueen tietämyksen perusteella ehtoja sille, milloin kaksi museoesinettä linkitetään toisiinsa.

Tietämuspohjaisen päättelyn pohjana voidaan käyttää aiemmissä menetelmissä kuvattuja tietoja, kuten esimerkiksi käyttäjäprofileja ja tietoa kohteista. Tässä mielessä tietämuspohjainen suosittelu yleistää aiemmin esitettyjä menetelmiä. Lisäksi päättelyyn voidaan käyttää yleisempää tietoa aihealueesta, kuten esimerkiksi tietoa oleellisista käsitteistä ja niiden välisistä suhteista (ontologiat).

Tietämuspohjaisessa suosittelussa käyttäjien tarpeet ja kohteet yhdistävät säännöt määritellään eksplisiittisesti. Muissa suosittelumenetelmissä suosittelusäännöt päätellään aineistosta etsimällä esimerkiksi yhdenmukaisia käyttäjäprofileja tai yhdenmukaisia dokumentteja. Verrattuna utiliteettipohjaiseen suositteluun, tietämuspohjaisessa suosittelussa päättelyketju käyttäjän tarpeiden ja kohteiden ominaisuuksista välillä on tyypillisesti pidempi ja monimutkaisempi.

Mikäli tietämuspohjaisen suosittelun säännöt tuotettaisiin tiedonlouhinnan (data mining) [HK01] menetelmillä, olisi ero tietämuspohjaisen suosittelun ja muiden suosittelumenetelmien välillä vähäinen. Tiedoneristämisen avulla voidaan esimerkiksi elintarvikeliikkeen ostosrekisteristä eristää sääntöjä siitä, mitkä ovat tyypilliset tuotteet, joita asiakkaat ostavat samalla kertaa.

2.8 Hybridisuosittelevjärjestelmät

Perinteiset suosittelevjärjestelmät tuottavat suosituksia perustuen yhteen tietotyypin, kuten esimerkiksi tietokanta käyttäjien antamista numeerisista arvioista kohteena oleville tuotteille. Tämän ratkaisun heikkous on se, että mikäli numeerisia arvioita ei ole, ei suosituksia pystytä tekemään. Ratkaisu ongelmaan on ns. hybridisuosittelevjärjestelmät, joissa yhdistetään kahden tai useamman suosittelevmenetelmän hyviä puolia ja pyritään näin välttämään kunkin yksittäisen menetelmän heikkouksia [Bur02]. Tyypillisessä hybridisuosittelevjärjestelmässä yhdistetään yhteistoiminnallinen suosittelev johonkin muuhun menetelmään, jotta vältettäisiin uuden käyttäjän ja uuden kohteen ongelmat.

Luvussa 2.4 esitelty Fab-suosittelevjärjestelmä sisältää myös hybriditoiminnallisuuden, jossa sisältöpohjaista suosittelevtä täydennetään yhteistoiminnallisella suosittelevulla [BS97]. Käyttäjän antamia pistearvioita hyödynnetään, paitsi agenttien opettamiseen, myös yhteistoiminnallisesti siten, että jos käyttäjä antaa tarpeeksi suuren pistemäärän tietylle WWW-sivulle, suositellaan kyseistä sivua käyttäjän mielipidenaapuruston jäsenille.

Fabin hybridiominaisuuksilla saavutetaan seuraavat edut [BS97]: Yhteistoiminnallisen suosittelevn ansiosta käyttäjille suositellaan myös niitä kohteita, jotka muiden käyttäjien mielestä ovat hyviä, mutta joita ei sisällön perusteella olisi suositeltu. Sisältöpohjainen suosittelev puolestaan mahdollistaa sellaisten kohteiden suosittelevn, joita kukaan muu käyttäjä ei ollut nähnyt. Sisältöpohjaisen suosittelevn ansiosta käyttäjälle voidaan tuottaa suosituksia myös siinä tilanteessa, että (yhteistoiminnalliselle suosittelevlle välttämättömiä) samanhenkisiä muita käyttäjiä ei löydy.

3 Monilähteiset suosittelujärjestelmät

Edellisessä luvussa kuvatut suosittelujärjestelmät ovat poikkeuksetta suljettuja järjestelmiä, jotka hyödyntävät suositteluun ainoastaan järjestelmän sisäisiä tietoja käyttäjistä ja kohteista. Itse kohteet voivat sinänsä olla peräisin järjestelmän ulkopuolelta, kuten esimerkiksi WWW-sivuja suosittlevan Fab-järjestelmän [BS97] tapauksessa. Lisäksi järjestelmissä voidaan hyödyntää tausta-aineistoja, kuten Kruulichin demografiapohjaisessa järjestelmässä [Kru97]. Suorituksen aikana järjestelmät eivät kuitenkaan hyödynnä ulkopuolista tietoa suositusten tuotannossa, josta seuraa se, että järjestelmien sisältämä tieto voi olla näkökulmaltaan rajoittunutta ja monipuolisuudeltaan suppeaa, eikä välttämättä sisällä ajantasaista tietoa kohteista, kuten esimerkiksi elokuvateattereiden kuluvan viikon ohjelmistoa.

Esimerkiksi elokuvateatterissa esitettävän elokuvan valintaan vaikuttaa paitsi elokuvan ”hyvyys”, myös missä ja mihin aikaan elokuvaa esitetään. Käyttäjän kannalta voi myös olla oleellista tietää elokuvaan ja esityspaikkaan liittyviä sellaisia tietoja, joita suosittelujärjestelmän tarjoaja ei ole tullut ajatelleeksi. Elokuvan ja elokuvateatterin valintaan voi esimerkiksi vaikuttaa se, pääseekö joukkoliikenneyhteyksillä oikeaan aikaan elokuvateatterille ja esityksen jälkeen sieltä pois, tai soveltuuko teatteri myös pyörätuolilla liikkuville. Tarkastelkaamme seuraavaa esimerkkikäyttötapausta:

Ville on etsimässä sopivaa elokuvaa illaksi. Elokuvan tulisi olla ”hyvä” ja sen tulisi kuulua toimintaelokuvien genreen ja olla sallittu alle 15-vuotiaille. Lisäksi elokuvateatterin pitäisi olla jokin alle kilometrin säteellä sijaitsevista teattereista, näytöksen pitäisi olla klo 17-23 välillä ja teatteriin pitäisi päästä pyörätuolilla.

Perinteisten suosittelujärjestelmien avulla Ville todennäköisesti löytäisi useita ”hyviä” elokuvia, mutta selvittääkseen, mitkä niistä pyörivät lähiseudun teattereissa,

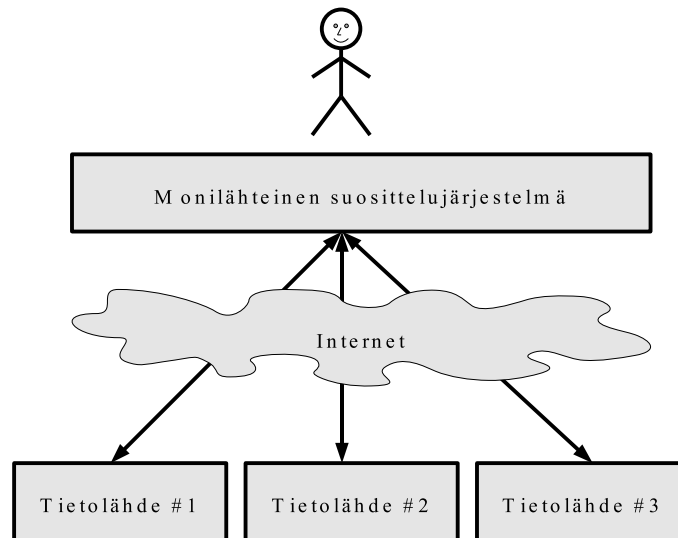
hän joutuisi hakemaan suosittelujärjestelmän lisäksi tietoa myös muista lähteistä. Olettaen, ettei Ville entuudestaan tiedä mitä elokuvateattereita on lähiseudulla, ensimmäiseksi hänen pitää selvittää, mitä teattereita on tarjolla annetun etäisyyden sisällä esimerkiksi Keltaiset sivut -tyyppisen palvelun avulla. Tämän jälkeen kunkin elokuvateatterin² Internet-sivujen kautta löytyy todennäköisesti elokuvien esitysajat yms tietoa. Elokuvateattereidenkaan sivuilta ei kuitenkaan välttämättä löydy kaikkea tietoa, etenkin sellaista, jonka julkaiseminen on epäedullista teatterille. Esimerkiksi esteettömyystietoja ei löytynyt Helsingin johtavien elokuvateattereiden Internet-sivuilta³. Tällaisen tiedon julkaiseminen olisi todennäköisesti jonkin kolmannen osapuolen, kuten sosiaali- ja terveystieteiden tai aiheeseen liittyvien järjestöjen, intresseissä.

Monilähteisten suosittelujärjestelmien (multi-source recommenders, meta recommenders) tavoitteena on ratkaista edellä kuvattu ongelma tarjoamalla käyttäjälle kaikki valinnan tekemisen kannalta oleelliset tiedot yhden käyttöliittymän avulla, joka huomioi sekä subjektiiviset mielipiteet että käyttäjän valinnan kannalta oleelliset muut rajoitteet hyödyntäen ja yhdistäen tietoa useista tietolähteistä (kts kuva 2) [Sch01, SKR02]. Monilähteisessä suosittelujärjestelmässä käyttäjän ei itse tarvitse hakea tietoa useista eri lähteistä ja asettaa kohteita paremmuusjärjestykseen, vaan tämä tapahtuu automaattisesti järjestelmän toimesta. Monilähteinen suosittelujärjestelmä parantaa suositusten käyttökelpoisuutta huomioimalla useita käyttäjän kannalta oleellisia näkökulmia, jotka voivat olla sekä subjektiivista ("hyvä" elokuva) että objektiivista tietoa (elokuvan esitysajat, ikäraja jne).

Monilähteiset suosittelujärjestelmät laajentavat luvussa 2.8 kuvattuja hybridisuosittelujärjestelmiä siten, että monilähteisissä suosittelujärjestelmissä hyödynnetään useiden suosittelumenetelmien lisäksi useista *autonomisista lähteistä* (esimerkiksi eri

²Esimerkiksi <http://www.finnkino.fi>

³<http://www.finnkino.fi> ja <http://www.kinopalatsi.fi> (3.7.2005)



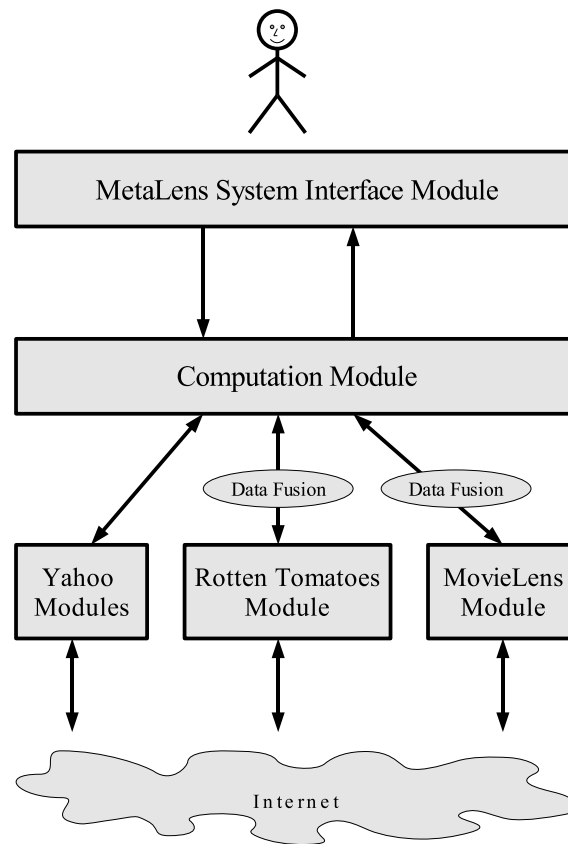
Kuva 2: Monilähteinen suosittelevjärjestelmä hyödyntää useita tietolähteitä suosituksen tuottamisessa käyttäjälle.

Internet-palveluista) kerättäviä tietoja.

3.1 MetaLens-järjestelmä

MetaLens on prototyyppi monilähteisestä suosittelevjärjestelmästä, jonka avulla käyttäjälle tuotetaan henkilökohtaisten valintojen mukaisia suosituksia perustuen useista eri tietolähteistä kerättyyn tietoon sekä hyödyntäen useita eri suosittelevmenetelmiä [Sch01]. MetaLens-järjestelmä hyödyntää kolmea tietolähdettä: MovieLens-järjestelmää [Mov], Yahoona [Yah] elokuvateatteritietoja sekä elokuva-arvosteluita sisältävää RottenTomatoes-palvelua [Rot].

MetaLens-järjestelmä koostuu kolmesta pääkomponentista (kuva 3): käyttöliittymästä (MetaLens System Interface Module), laskentamoduulista (Computation Module) ja tietolähdekohtaisista tiedonkeruumoduuleista (esimerkiksi Yahoo Module). Seuraavassa tarkastellaan tarkemmin kunkin osan toimintaa.



Kuva 3: MetaLens-järjestelmä [Sch01]

3.1.1 Käyttöliittymä

MetaLensing hakukäyttöliittymän (kuva 4) avulla käyttäjä voi antaa elokuvan valintaan liittyviä ehtoja ja toiveita, kuten genre-valinta ja etäisyys teatteriin. Lisäksi kutakin valintaehtoa voidaan painottaa “merkityksettömästi” (not important) “pakolliseen ehtoon” (must have). Haun tuloksena MetaLens tuottaa paremmuusjärjestyksessä olevan listan elokuvanäytöksistä, jotka parhaiten vastaavat käyttäjän asettamia ehtoja.

MetaLens Preference Screen - Microsoft Internet Explorer

File Edit View Favorites Tools Help

What Movie Should I see Tonight?

This feature allows you to receive recommendations based on the movies actually showing in your area and based on your current mood and preferences. Edit your preferences and rate the features' importance to you. Changing the settings should change the order of the recommendation results.

Zip Code to use

| Not Important | Very Important | Must Have | Movie Features | Preferences |
|-----------------------|----------------------------------|----------------------------------|-----------------------------------|---|
| <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Genre(s) | <input type="checkbox"/> Action <input type="checkbox"/> Art <input checked="" type="checkbox"/> Comedy <input type="checkbox"/> Documentary <input type="checkbox"/> Drama <input checked="" type="checkbox"/> Family <input type="checkbox"/> Horror <input checked="" type="checkbox"/> Musicals <input type="checkbox"/> Romance <input type="checkbox"/> SciFi <input type="checkbox"/> Thriller |
| <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | MPAA Rating(s) | <input checked="" type="checkbox"/> G <input checked="" type="checkbox"/> PG <input checked="" type="checkbox"/> PG-13 <input type="checkbox"/> R <input type="checkbox"/> NC-17 <input type="checkbox"/> NR |
| <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | Film Length | At least <input type="text" value="60"/> minutes. |
| <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Film Length | Not longer than <input type="text" value="120"/> minutes. |
| <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Objectionable Content | Should not contain <input checked="" type="checkbox"/> Violence <input type="checkbox"/> Sensuality <input checked="" type="checkbox"/> Crude Humor <input checked="" type="checkbox"/> Sex <input type="checkbox"/> Terror <input type="checkbox"/> Drug Use |
| <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Critic's Rating | |
| <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | MovieLens Personalized Prediction | |
| Not Important | Very Important | Must Have | Theater Features | Preferences |
| <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Distance to Theater | No more than <input type="text" value="10"/> miles |
| <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | Start Time | Not before: <input type="text" value="7:00 PM"/> |
| <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | End Time | Not after: <input type="text" value="9:00 PM"/> |

Done Internet

Kuva 4: MetaLens-järjestelmän hakukäyttöliittymä [Sch01]

3.1.2 Tiedonkeruumoduulit ja tiedon fuusiointi

Yahoon tiedot elokuvista ja elokuvateattereista ovat MetaLensin ensisijaisia tietoja, joihin MovieLensin ja RottenTomatoesin tiedot yhdistetään eli fuusioidaan (data fusion). MovieLens tarjoaa käyttäjille henkilökohtaisia suosituksia perustuen heidän arvioihinsa aiemmin katsomistaan elokuvista yhteistoiminnallista suosittelua hyödyntäen. MovieLensin avulla käyttäjä voi löytää ”kiinnostavat” elokuvat, mutta muita tietoja elokuvista ei juurikaan ole. RottenTomatoes-palvelusta kullekin

elokuvalle saadaan arvostelutietoina mm. tiedot siitä, millaisen (numero)arvostelun elokuva on saanut ja kuinka monta arvostelua elokuva on saanut. Tietomoduulit ovat käytännössä kääresovelluksia (wrapper), joiden avulla tietolähteinä käytettävistä HTML-sivuista jäsennetään tarvittavat tiedot.

Tietofuusio-vaiheessa nämä eri lähteistä kerätyt tiedot yhdistetään toisiinsa siten, että ne muodostavat yhtenäisen tietokannan, josta voidaan esimerkiksi saada selville kaikki tiettyyn elokuvaan liittyvät tiedot riippumatta siitä, mistä lähteestä mikäkin tieto on alunperin kerätty. Tietofuusio tehdään käyttäen sellaisia yksilöiviä tietoja, jotka ovat pääsääntöisesti samoja fuusioitavien tietolähteiden välillä. MetaLensissä elokuvan nimi ja julkaisuvuosi toimivat avaimena, jonka avulla yhdistetään samaan elokuvaan liittyvät eri lähteistä saadut tiedot. Elokuviin yksilöinti nimen avulla ei kuitenkaan ole ongelmattonta, koska esimerkiksi eri lähteissä käytetään eri tapoja kirjoittaa elokuvan nimi (“The Thomas Crown Affair” vs “Thomas Crown Affair, The”). Osumatarkkuuden parantamiseksi MetaLensissä käsitellään merkkijonoja siten, että niistä poistetaan usein toistuvia sanoja (esimerkiksi “and”, “or” ja “the”) sekä välimerkkejä, kuten pisteitä ja pilkkuja. Virheiden välttämiseksi ylläpitäjä tarkistaa vielä lopuksi kaikki uudet elokuvanimiin perustuvat fuusiot kullakin viikolla. Käsintehdyt tarkistukset on mahdollista tehdä, koska uusia elokuvia ilmestyy viikottain kuitenkin suhteellisen vähän, joten työmäärä pysyy kohtuullisena.

3.1.3 Laskentamoduuli

Laskentamoduulin tärkein tehtävä on järjestää tarjolla olevat elokuvaesitykset paremmuusjärjestykseen käyttäjän antamien kriteereiden mukaan. Paremmuusjärjestys muodostetaan käyttäen Saltonin et al. laajennettujen Boolean-hakujen similariteettimittaa [SFW83], joka laajentaa perinteistä Boolean-kyselyä siten, että myös osittaiset osumat sallitaan. Esimerkiksi jos käyttäjä etsii elokuvia hakuehdolla “elokuvat, joiden pituus on alle 130 minuuttia JA elokuvan genre on komedia”, hakutu-

lokseen palautetaan täysien osumien lisäksi myös pienemmällä prioriteetilla ne elokuvat, joilla jompi kumpi ehdoista on voimassa. Lisäksi Saltonin similariteettimitta mahdollistaa hakuehtojen painotuksen.

MetaLensissä käytettävä Boolean-similariteetin laskentakaava on seuraava:

$$\text{sim}(I, Q) = 1 - \sqrt{\frac{\sum_{a \in \text{ehdot}} w_a^2 (1 - d_a)^2}{\sum_{a \in \text{ehdot}} w_a^2}} \quad (3)$$

Kaavassa I on arvioitava kohde (elokuvaesitys), jota vertaillaan kyselyyn Q . Käyttäjän antamien hakuehtojen osuvuus kohteeseen I lasketaan yhteen siten, että a on kulloinkin käsiteltävä ehto, w_a on kyseiselle ehdolle annettava paino (arvoalue $[0, 1]$) ja d_a on ehtokohtainen etäisyysmitta, jonka arvon määräytyminen jakautuu kolmeen tapaukseen:

Yksittäisien ehtojen (“elokuvan pituus on alle 130 minuuttia”) osalta d_a saa arvokseen yksi mikäli ehto toteutuu, nolla jos ei toteudu.

Useita aliehtoja sisältävät ehdot, kuten esimerkiksi genre-luokitus, arvotetaan siten, että d_a saa arvokseen yksi mikäli yksikin ehto I :llä täyttyy. Muussa tapauksessa arvoksi tulee nolla. Schafer toteaa (tarkemmin perustelematta), että elokuvien etsijän kannalta ei hänen mielestään ole eroa sen suhteen, osuuko useampi kuin yksi vaihtoehtoisista arvoista kohteeseen I ja tämän vuoksi arvo on aina joko nolla tai yksi [Sch01]. Näin ollen, esimerkiksi kohteet, joilla genre on “komedia”, “musikaali” tai “komedia JA musikaali” arvostetaan yhtä hyväksi hakutuloksiksi MetaLensissä. Päinvastoin kuin Schafer, Salton et al. ovat puolestaan päätyneet siihen, että usein kohde, jossa on useampi OR-ehto voimassa, on mielekästä arvostaa paremmaksi osumaksi kuin vähemmän OR-osumia sisältävät kohteet [SFW83]. Jää epäselväksi miksei näin toimita myös MetaLensissä.

Kolmas tapaus d_a :n arvon määrittelyksi on *numeeristen arvioiden* yhdistäminen ha-

kuun. Numeerisia arvioita ovat esimerkiksi MovieLensin tuottamat käyttäjän mukaan personoidut pisteet elokuville (tähdet yhdestä viiteen). Näissä tapauksissa d_a saa arvokseen normalisoidun pistemäärän (arvoalue $[0, 1]$). Esimerkiksi MovieLens-tähdet normalisoituvat siten, että yksi tähti vastaa arvoa nolla, kolme tähteä arvoa 0,5 ja viisi tähteä arvoa yksi.

3.1.4 Hakutulosten esittäminen

MetaLensin hakutulostilaus on taulukko suosituksista, jossa kullakin rivillä on yksi elokuvaesitys ja sarakkeissa erilaisia tietoja elokuvaan ja elokuvaesitykseen liittyen. MetaLens-tutkimuksen aikana kokeiltiin useita eri vaihtoehtoja sille, mitä tietoja taulukon sarakkeissa esitettiin ja missä määrin käyttäjä pystyi vaikuttamaan sarakkeiden sisältöön.

Suppeimmassa versiossa hakutuloksessa esitettiin ainoastaan suosituksen MetaLens-pistemäärä, elokuvan nimi, elokuvateatterin nimi ja esityksen alkamisaika. Hieman laajemmassa hakulistauksessa esitettiin kahdeksan tietoa, joissa edellisten lisäksi oli MovieLens-pistemäärä, elokuvan genre, etäisyys teatteriin ja kriitikoiden antama pistemäärä.

Laajimmassa versiossa oli 16 tietoa, joissa edellisten lisäksi esitettiin mm. kävijöiden keskimääräinen arvosana elokuvalla ja teatterin palvelut vammaisille. Lisäksi MetaLensissä kokeiltiin myös räätälöityjä tuloslistauksia, joiden sisältö määräytyi dynaamisesti. Yksi versio toimi siten, että käyttäjä sai valita, mitä tietoja tuloslistauksessa esitettiin. Toisessa versiossa tuloslistan tiedot määräytyivät automaattisesti siten, että listauksessa näytettiin ne tiedot, joita käyttäjä painotti eniten haussa.

Käyttäjätesteissä koehenkilöt pitivät edellä kuvatuista näkymistä eniten manuaalisesti räätälöidystä tulostuloksesta. Toiseksi parhaimmaksi arvioitiin automaattisesti räätälöity tuloslista sekä kahdeksan tietoa sisältävä vaihtoehto. Paljon tietoa

sisältävä 16 tiedon listaus ja pelkät perustiedot sisältävä listaus arvioitiin huonoimmiksi.

Edellisen pohjalta todettiin, että käyttäjät pitävät hyödyllisenä lisätietojen saamista elokuvia hakiessaan, vahvistaen näin käsitystä monilähteen suosittelujärjestelmän perusidean mielekkyydestä. Lisäksi todettiin, että suurempien tietomäärien yhteydessä (MetaLensin tapauksessa esimerkiksi 16 tietoa) käyttäjät arvostavat mahdollisuutta rajoittaa esitettävän tiedon määrää.

3.1.5 Johtopäätökset

MetaLens-tutkimuksen puitteissa todettiin käyttäjien pitävän monilähteisiä suosittelujärjestelmiä hyödyllisinä ja usein käyttökelpoisempina kuin perinteisiä suosittelujärjestelmiä. Käyttäjät arvostavat myös sitä, että järjestelmän toimintaa voi räätälöidä sekä hakua tehdessä (esimerkiksi painotukset) että tulostan osalta.

Jatkotutkittavaa MetaLensissä olisi tutkimuksen mukaan mm. seuraavasti: Miten suosittelutietoja voitaisiin kytkeä joustavammin osaksi monilähteistä suosittelujärjestelmää siten, että käsityötä olisi vähemmän ja mahdollisesti jopa niin, että käyttäjä voisi lisätä omia lähteitään suositusten osaksi? Miten tulosten järjestelyalgoritmia voisi jatkokehittää, jotta parhaimmat suositukset tulisivat ensimmäiseksi? Miten tietoja käyttäjälle pitäisi tarjota tulossivulla? Miten tulossivujen automaattinen räätälöinti käyttäjän tarpeiden mukaan saataisiin toimivaksi?

3.2 Monilähteen suosittelun haasteet

Edellä esitetyn perusteella ja edellä esitettyjä ideoita kehittämällä seuraavassa luetellaan muutamia mielenkiintoisia haasteita monilähteen suositteluun liittyen, joita tutkielman loppuosassa pyritään ratkaisemaan semanttisen webin keinoin.

Haaste 1. Miten autonomisia tietolähteitä hyödynnetään suosittelussa?

MetaLensin jatkokehityskohteenä mainittiin se, miten ulkopuolisia tietolähteitä voitaisiin aiempaa helpommin kytkeä osaksi monilähteistä suosittelujärjestelmää. Optimitilanteessa ulkopuolisen tiedon käyttöönotto tapahtuisi täysin automaattisesti ja jopa siten, että käyttäjä voisi antaa haluamiaan lähteitä (esimerkiksi URL-osoite) käytettäväksi suositusten tuotannossa.

Haaste 2. Miten eri lähteistä tulevat sisällöt yhdistetään?

MetaLensin tietofuusio tapahtuu puoliautomaattisesti siten, että elokuvia täsmäytetään toisiinsa niiden otsikon ja julkaisuvuoden avulla. Tämän jälkeen ylläpitäjä vielä tarkasti joka viikko osumien oikeellisuuden. Tietolähteitä olisi joustavinta käyttää, mikäli tiedot yhdistyisivät automaattisesti, ilman ylläpitäjän valvontaa.

Haaste 3. Miten suositukset tuotetaan monilähteistä tietoa hyödyntäen?

MetaLensin tapauksessa suositukset tuotetaan yhdistämällä eri tietoja toisiinsa Saltonin laajennetulla Boolean-hakualgoritmilla, mutta samalla jätettiin avoimeksi kysymys siitä, mitä muita algoritmeja monilähteisissä suosittelujärjestelmissä voisi käyttää. Monilähteisen suosittelujärjestelmän monikäyttöisyyttä ja modulaarisuutta parantaisi se, että tietovarastoa voitaisiin käsitellä useilla eri algoritmeilla esimerkiksi aihealueesta tai käyttäjästä riippuen. Mielekkäitä algoritmeja käytettäväksi voisivat olla myös esimerkiksi perinteisten suosittelujärjestelmien (luku 2) menetelmät riippuen siitä, millaista tietoa monilähteiseen suosittelujärjestelmään on saatavilla.

Haaste 4. Miten monilähteiset suositukset esitetään käyttäjälle?

MetaLens-järjestelmässä tutkittiin, mitä tietoja käyttäjälle pitäisi suosituksiin liittyen esittää ja missä muodossa. Ratkaisuna oli tulostaulukko, johon valittiin joukko

tietoja kuhunkin elokuvaan liittyen. Mitä enemmän ja monimuotoisempaa tietoa on käytettävissä, sitä monipuolisemmin tietoa voidaan halutessa näyttää myös käyttäjälle suosittelutulosten yhteydessä.

4 Semanttinen web

4.1 WWW:n sovelluskäytön ongelmat

Monilähteisten suosittelujärjestelmien ideana on parantaa suosittelujärjestelmien laatua hyödyntämällä useita tietolähteitä, kuten WWW-sivuja, suositusten tuottamisessa ja tarjoamalla käyttäjälle kerralla kaikki valinnan tekemisen kannalta oleellinen tieto. Tällä hetkellä WWW-sisällön hyödyntäminen ohjelmallisesti on kuitenkin hankalaa, koska sisältö on esitetty pelkästään ihmisen tulkittavaksi tarkoitettussa muodossa, kuten HTML- [HTML] tai PDF-muodossa [PDF].

Heikosta koneluettavuudesta johtuen, esimerkiksi luvussa 3.1 kuvatussa MetaLens-järjestelmässä käytettiin tietojen keräämiseen Internet-sivustoilta ns. kääresovelluksia (wrapper) eli käsin tehtyjä ohjelmia, joilla erotettiin HTML-sivun sisältö ulkoasusta perustuen sivujen ulkoasulliseen säännönmukaisuuteen. Kääresovellukset perustuvat olettamukseen siitä, että sivustojen ulkoasu pysyy muuttumattomana ja että tietyt tiedot esitetään aina samalla tavalla. Esimerkiksi elokuvan otsikko ja vuosiluku esitetään Internet Movie Database (IMDB) -elokuvatietokannassa seuraavasti⁴:

```
<h1><strong class="title">Star Wars
<small>(<a href="/Sections/Years/1977">1977</a></small></strong></h1>
```

Ihmisen on helppo päätellä ensinnäkin se, että kyse on elokuvasta, koska kuvattu kohde on haettu elokuvatietokannasta ja kyse on varsin yleisesti tunnetusta elokuvasta. Esimerkistä voi myös päätellä, että "Star Wars" on elokuvan otsikko, "1977" on julkaisuvuosi ja että elokuvan otsikko ja julkaisuvuosi liittyvät samaan elokuvaan. Ohjelmallisesti edellisen tiedon päättely on vaikeampaa, koska tiedon esittämiseen ei ole käytetty mitään yleisesti sovittua tiedonesityskieltä tai terminologiaa. Koneel-

⁴esimerkkisivu: <http://www.imdb.com/title/tt0076759/>

lisesti voidaan lähinnä tulkita, että kyse on HTML:n h1-tyyppisestä otsikosta, jonka sisällä on erilaisia muotoiluja ja yksi linkki.

Toinen ongelma WWW-sivustojen hyödyntämisessä on se, että eri sivustoilla käytetään eri tunnisteita samoille kohteille, kuten elokuville, jolloin ongelmana on tunnistaa, mitkä kohteet ovat eri järjestelmissä samoja ja mitkä eivät. Eräillä kohteilla, kuten kirjoilla ja CD-levyillä, on kansainvälisiä yksilöintitunnisteita kuten The International Standard Book Number (ISBN) [ISB] kirjoilla ja The International Standard Recording Code (ISRC) [ISR] musiikkitalenteilla. Sen sijaan esimerkiksi elokuvilla ei ole kansainvälistä yksilöintikoodia. Jos yhtenäistä koodia ei ole käytettävissä, tietojen yhdistäminen vaatii yleensä erilaisia heuristiikkoja ja mahdollisesti käsityötä. Esimerkiksi MetaLens-tapauksessa Yahoo, Rotten tomatoes ja MovieLens-järjestelmät käyttävät kukin erilaisia sisäisiä tunnisteita elokuville. Tästä johtuen elokuvat yhdistetään MetaLensissä elokuvan otsikon ja julkaisuvuoden perusteella. Mahdolliset ongelmatapaukset ohjataan ylläpitäjän ratkaistavaksi.

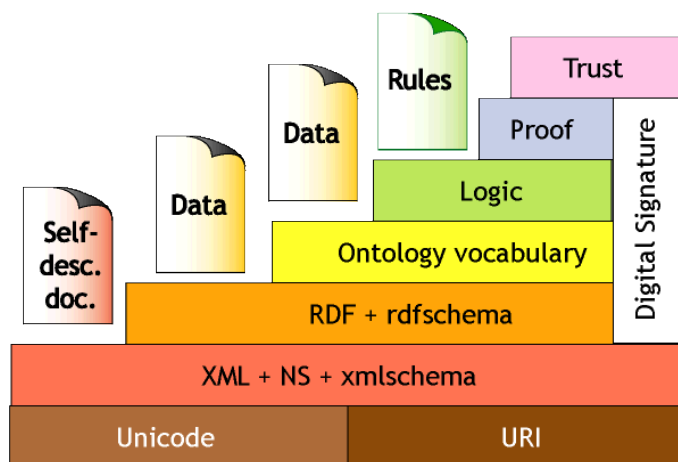
Kolmas ongelma nykyisessä WWW:ssä on se, että relaatioiden kuvaamiseen sivujen välillä on ainoastaan anonyymejä ``-linkkejä, jotka eivät sisällä mitään tietoa siitä, *miksi* kaksi sivua liittyvät toisiinsa. Ihmisen voi olla helppo tulkita, miten linkitetyt sivut liittyvät toisiinsa (esimerkiksi sivun A tekijä esitellään sivulla B tai sivu B on vanhempi versio sivusta A), mutta ohjelmallisesti tämä on hankalaa. Sama ongelma koskee myös sivun sisäisten elementtien suhteita kuten esimerkiksi elokuvan otsikon ja julkaisuvuoden liittymistä toisiinsa. Anonyymejä linkkejä sivun sisällä on mahdollista tehdä `` />-rakenteella, mutta tämä ei kerro miten linkitetyt kohteet liittyvät toisiinsa.

Edellä kuvatuista ongelmista seuraa se, että nykyinen web soveltuu huonosti sovelusten käytettäväksi.

4.2 Semanttisen webin tekniikat

WWW:n sovelluskäytön kehittämiseksi WWW-konsortio (W3C) on käynnistänyt hankkeen, jossa tavoitteena on luoda seuraavan sukupolven web, *semanttinen web* (Semantic Web) [BLHL01, Hyv02, AvH04, SW]. Semanttisen webin ideana on kuvata sisältö käyttäen eksplisiittisesti määriteltyjä yhteisiä käsitteitä, jolloin sisältöä voidaan prosessoida ohjelmallisesti kyseisiä käsitteitä ja niiden välisiä suhteita hyödyntäen [Usc03].

Semanttisella webillä pyritään parantamaan tiedon löydettävyyttä ja luomaan uusia navigointimahdollisuuksia WWW- ja muihin sisältöihin. Lisäksi tavoitteena on helpottaa eri lähteistä koostuvan tiedon hyödyntämistä ja tähän perustuvaa aiempaa monimutkaisempien tehtävien automatisointia [KM02]. Esimerkkinä sovellusongelmasta, johon semanttinen web tarjoaisi automaatiomahdollisuuksia, on ajanvarauksen tekeminen lääkäriin, jossa sovellus voisi semanttisen webin tekniikoiden avulla vertailla potilaan ja lääkärin kalenteria ja etsiä kummallekin sopivan tapaamisaajan [BLHL01].



Kuva 5: Semanttisen webin arkkitehtuuri [BL00].

Semanttinen web muodostuu useista eri tekniikoista sekä yhteisesti sovituisista toi-

mintaperiaatteista. Kuvassa 5 on “webin isän” Tim Berners-Leen vuonna 2000 esittämä näkemys semanttisen webin arkkitehtuurista. Arkkitehtuuri rakentuu olemassa olevien WWW-tekniikoiden, kuten URI-osoitteiden ja XML-kielen, päälle, jotta yhteensopivuus nykyiseen WWW:hen säilyisi. Seuraavassa käydään tarkemmin läpi arkkitehtuurin kerrokset alhaalta ylöspäin lukien.

4.2.1 Resurssien yksilöinti: URI

Semanttinen web perustuu Uniform Resource Identifier (URI) [URIA] -tunnisteiden käyttöön kuten perinteinenkin web (URL on myös URI [URIB]). Tunnisteiden avulla yksilöidään kaikki semanttiseen webiin kuuluvat kohteet (resurssit), jotka voivat olla esimerkiksi WWW-sivuja, mutta myös tietokoneen ulkopuolisia asioita, kuten henkilöitä, paikkoja, elokuvia ja käsitteitä. Esimerkiksi `http://www.cs.helsinki.fi` on Helsingin yliopiston Tietojenkäsittelytieteen laitoksen URI, tämän tutkielman kirjoittajan URI on `mailto:kim.viljanen@iki.fi` ja `owl:Thing` on URI käsitteelle, joka kuvaa kaikkien käsitteiden yläluokkaa. Esimerkissä `owl:` viittaa nimiavaruuteen (namespace), joka määrittää URI:na. Sen arvo `owl:`-tapauksessa on `http://www.w3.org/2002/07/owl#`. Nimiavaruuden perään katenoidaan esimerkiksi tapauksessa `Thing`, jolloin muodostuu käsitteen todellinen URI.

Vaikka URI:n ei välttämättä tarvitse osoittaa mihinkään olemassa olevaan verkkoosoitteeseen (toisin kuin URL), hyvänä käytäntönä pidetään sitä, että URI:n domainosio (esimerkiksi `cs.helsinki.fi`) on sen tahon hallinnassa, joka ottaa käyttöön kyseiseen domainiin kuuluvia osoitteita. Samalla kohteella voi olla myös enemmän kuin yksi URI, kuten esimerkiksi henkilöllä, jolla on useita sähköpostiosoitteita.

4.2.2 Resurssien välinen verkko: RDF

Nykyisessä webissä <a href...>-linkit eivät sisällä tietoa siitä, mikä suhde kahden linkitetyn kohteen välillä on. Semanttisessa webissä puolestaan URI:en väliset linkit voidaan tyypittää kolmannen URI:n avulla, joka kertoo linkin tyyppin. Linkitysten tekemiseen W3C suosittelee RDF-tietomallia (Resource Description Framework) [RDFa], joka sisältää tarvittavat välineet tietojen esittämiseen semanttisessa webissä.

RDF-kolmikko koostuu subjektista, predikaatista ja objektista, jossa subjekti ja predikaatti ovat osoitteita ja objekti voi olla joko URI tai merkkijono. RDF-kolmikoiden avulla on mahdollista ilmaista metatietoa mistä tahansa subjekteista. Esimerkiksi tieto siitä, että *Helsinki on osa Uudenmaan maakuntaa* voidaan kertoa esimerkiksi seuraavasti ns-esimerkinimiavaruudessa⁵:

```
(subjekti)   (predikaatti)   (objekti)
ns:Helsinki ns:kokonaisuus ns:Uusimaa.
```

Tieto siitä, että Helsingin nimi on "Helsinki" voidaan puolestaan esittää seuraavasti:

```
ns:Helsinki rdfs:label "Helsinki".
```

Edellä kuvatulla tavalla voidaan muodostaa URI-tunnisteiden avulla verkkomuotoinen tietorakenne, joka on samalla tavalla maailmanlaajuinen kuin esimerkiksi WWW. Lisäksi RDF:n tärkeä ominaisuus on se, että mikäli kaksi RDF-verkkoa sisältää vähintään yhden yhteisen URI:n, kytkeytyvät verkot yhteisten URI:en ansiosta toisiinsa. Luonnollisesti eri lähteistä tulevien RDF-verkkojen yhdistäminen voi aikaansaada ristiriitoja, johon liittyvää ongelmaa käsitellään jäljempänä.

RDF voidaan serialisoida useilla eri tavoilla. Tyypillisin on XML-pohjainen serialisaatio (katso liite 2), joten tästä johtuen RDF sijaitsee Berners-Leen arkkitehtuu-

⁵Esimerkeissä käytetään N3-syntaksia [N3].

rikaaviossa XML-kerroksen yläpuolella. Toinen yleisesti käytetty RDF-serialisaatio on Notation 3 (N3) -notaatio [N3], jota pidetään XML-serialisaatiota helppolukuisempaan.

4.2.3 RDF-skeemat ja ontologiat

Semanttisen webin tärkeimpiä ideoita on ajatus yhteisistä, jaetuista käsitteistä, joiden avulla tietoa kuvaillaan. Tämä jaettu käsitteistö määritellään ontologioiden (ontology) avulla. Yleisin määritelmä tietotekniikassa käytettäville ontologioille on: *Ontologia on formaali, eksplisiittinen määrittely yhteisestä käsitteistöstä*⁶ [Gru93].

Ontologian avulla kuvataan jonkin aihealueen ja käyttötarkoituksen kannalta oleelliset käsitteet ja niiden väliset suhteet, jolloin näitä voidaan käyttää sisällön esittämiseen tietojärjestelmissä kuten semanttisessa webissä. Jokainen tietojärjestelmä, ohjelmisto tai käyttäjä sitoutuu eksplisiittisesti tai implisiittisesti väistämättä aina johonkin käsitteistöön, jonka avulla tietoa esitetään. Ontologioiden ideana on tehdä tämä sidos eksplisiittiseksi, jolloin kaikki tiettyyn ontologiaan sitoutuneet tahot voivat ymmärtää toisiaan.

RDF Schema (RDFS) [RDFb] ja Web Ontology Language (OWL) [OWL] ovat molemmat W3C:n suosittelimia kieliä ontologioiden kuvaamiseen. Näistä RDFS on vanhempi ja tarkoitettu ainoastaan yksinkertaisten ontologioiden kuvaamiseen. OWL sisältää RDFS:ään verrattuna monipuolisempia ja tarkempia tapoja määritellä luokkia ja suhteita. Lisäksi OWL:ssa (erityisesti OWL DL⁷ -osakieli) on kiinnitetty huomiota ontologioiden tehokkuuteen, kun niitä hyödynnetään päättelykoneissa.

Esimerkiksi paikkoihin liittyvän tiedon kuvaamiseen voidaan määritellä ontologia, josta ilmenee aihealueeseen liittyvät käsitteet (luokat), kuten “Valtio”, “Kunta” ja “Joki”. Kuvassa 6 on esimerkki MuseoSuomessa (luku 5) käytettävästä paikkaonto-

⁶“An Ontology is an explicit specification of a conceptualization.”

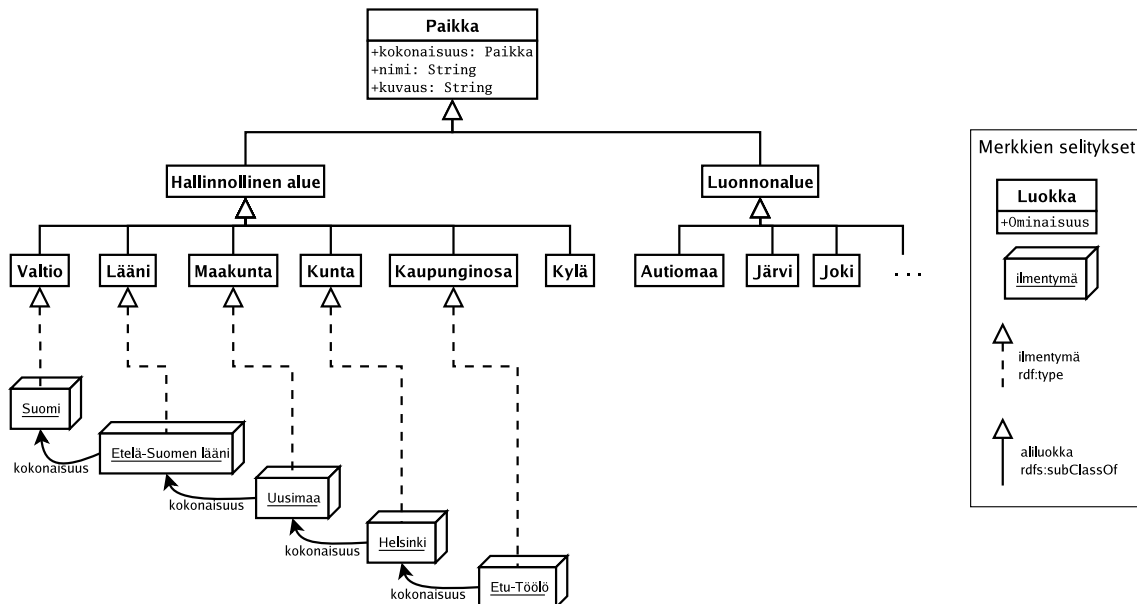
⁷DL tulee sanoista “Description Logic”.

logiasta, johon on mallinnettu paikkojen kuvaamiseen tarvittavia käsitteitä (luokkia) sekä kuvattu paikkojen suhteita toisiinsa (esimerkiksi Helsinki on osa Suomea) näitä käsitteitä hyödyntäen.

Ontologian avulla määritellään myös, mitä ominaisuuksia luokan ilmentymällä voi olla. Esimerkkiontologiassa on määritelty, että kaikkiin paikkoihin liittyy tieto siitä, minkä osa paikka on (“kokonaisuus”). Lisäksi jokaisella paikalla on merkkijonomuotoiset ominaisuudet “nimi” ja “kuvaus”. Ominaisuudet periytyvät aliluokille.

Ontologiat ja niiden ilmentymät voidaan esittää RDF- ja sitä täydentävillä muilla kielillä, kuten OWL:lla. Esimerkiksi tieto siitä, että *Helsinki* on *Kunta*-luokan ilmentymä esitetään seuraavasti:

```
ns:Helsinki rdf:type ns:Kunta.
```



Kuva 6: Esimerkki MuseoSuomen paikkaontologiasta.

Yksi suurimmista haasteista ontologioiden kehittämisessä on se, että usein eri tahoilla on osittain poikkeava käsitys yhteisistä käsitteistä. Tästä johtuen tarvitaan hyvin yleisiä ontologioita (yläontologioita), joissa on kuvattu peruskäsitteitä, joista

useimmat voivat olla samaa mieltä. Tällaisia yläontologioita ovat esimerkiksi Suggested Upper Merged Ontology (SUMO) [SUM] sekä Yleinen Suomalainen Ontologia (YSO) [YSO], joista jälkimmäistä kehitetään parhaillaan. Yläontologioiden käsitteitä laajentamalla voidaan luoda aihe- ja alakohtaisia ontologioita, jotka kytkeytyvät toisiinsa yläontologian käsitteiden kautta.

Ontologioiden luomiseen tarvitaan käytännössä editointisovellusta. Yksi tunnetuimmista graafisista työkaluista ontologioiden tuottamiseen on Protégé-editori, joka tukee mm. RDFS- ja OWL-ontologioiden tekemistä sekä sisältää useita erilaisia RDF-aineiston visualisointimahdollisuuksia [Pro].

4.2.4 Loogiset säännöt

Kuvan 5 logiikkakerros (logic) viittaa keinoihin kuvata loogisia sääntöjä, jotka hyödyntävät alla olevien ontologioiden käsitteitä ja niiden avulla ilmaistua tietoa päätelyiden tekemiseen. Käytännössä tällaista päättelyä voidaan tehdä esimerkiksi logiikkaohjelmointikieli Prologilla. Yksi Prologin toteutus on SWI-Prolog, joka sisältää tuen RDF:n käsittelylle [WSW03, SWI]. RDF-aineisto esitetään SWI-Prologissa siten, että jokainen RDF-kolmikko muodostaa faktan $rdf(s, p, o)$, jossa s on subjekti, p on predikaatti ja o on objekti. RDF-faktoja voidaan tämän jälkeen hyödyntää kuten mitä tahansa muuta Prologiin tallennettuja faktoja.

Esimerkiksi edellä esitetyt tiedot Helsinkiin liittyen näyttävät SWI-Prologiin luetuna seuraavalta:

```

rdf(ns:'Helsinki', ns:'kokonaisuus', ns:'Uusimaa').
rdf(ns:'Helsinki', rdfs:'label', literal('Helsinki')).
rdf(ns:'Helsinki', rdf:'type', ns:'Kunta').

```

Prolog-säännöt voivat sisältää mielivaltaisen monimutkaisia ehtoja. Jos kaikki säännön ehdot toteutuvat, evaluoituu sääntö todeksi. Esimerkiksi seuraavalla Prolog-säännöllä voidaan päätellä, mitkä ovat tiettyyn maakuntaan kuuluvien kuntien ni-

met:

```
osienNimet(Maakunta, Nimi) :-
  rdf(Osa, ns:'kokonaisuus', Maakunta),
  rdf(Osa, rdf:'type', ns:'Kunta'),
  rdf(Osa, rdfs:'label', literal(Nimi)).
```

Jos sääntöä kutsuu siten, että muuttujan `Maakunta` arvoksi asetetaan `ns:'Uusimaa'`, evaluoituu sääntö todeksi siten, että `Nimi`-muuttujan arvo on `'Helsinki'`. (Prologissa muuttujien nimet alkavat aina isolla kirjaimella ja vakiot pienellä kirjaimella.)

SWI-Prolog on osoittautunut tehokkaaksi ja käteväksi tavaksi käsitellä RDF-aineistoja. Järjestelmän kehittäjät ovat arvioineet kokeidensa perusteella, että SWI-Prologilla on mahdollista käsitellä keskusmuistissa jopa 40 miljoonaa RDF-kolmikkoa [WSW03]. Loogisten sääntöjen ohella RDF-aineistoa voidaan käsitellä myös proseduraalisesti. Tunnetuimpia Javalla tehtyjä RDF-työkaluja on HP Labsin Jena-kehysjärjestelmä, joka sisältää mm. ohjelmointirajapinnan RDF-aineiston käsittelyyn, tuen RDF-aineiston tallentamiseen relaatiotietokantaan sekä oman sääntökielen, jolla voi tehdä päättelyä hieman samaan tapaan kuin edellä Prologilla [Jen].

4.2.5 Todistukset

Kuvan 5 arkkitehtuurikaaviossa todistukset (proof) viittaavat siihen, millä keinoin tiedon oikeellisuus voidaan osoittaa esimerkiksi tilanteissa, joissa tieto on päätelty hyödyntäen eri puolilta kerättyjä semanttisen webin tietoja. Berners-Lee haluaa (Antioniounin ja van Harmelenin mukaan) semanttisen webin sovelluksiin “Oh yeah?”-napin, joka perustelisi haluttaessa käyttäjälle esitettyjä semanttisen webin avulla kerättyjä tietoja ja tehtyjä päätelmiä [AvH04].

4.2.6 Luottamus

Berners-Leen arkkitehtuurikaavion ylimpänä kerroksena on luottamus (trust), jolla viitataan tekniikoihin, joilla arvioidaan semanttisen webin tietojen sekä näistä

pääteltyjen tietojen luotettavuutta.

Tiedon luottamukseen liittyy ensinnäkin se, että voidaan luottaa siihen, että tiedon tuottaja voidaan tunnistaa. Tämä voidaan varmistaa esimerkiksi digitaalisella allekirjoituksella, jolloin voidaan varmistaa tiedon alkuperä ja huomata mahdolliset yritykset muuttaa tietoa luvatta.

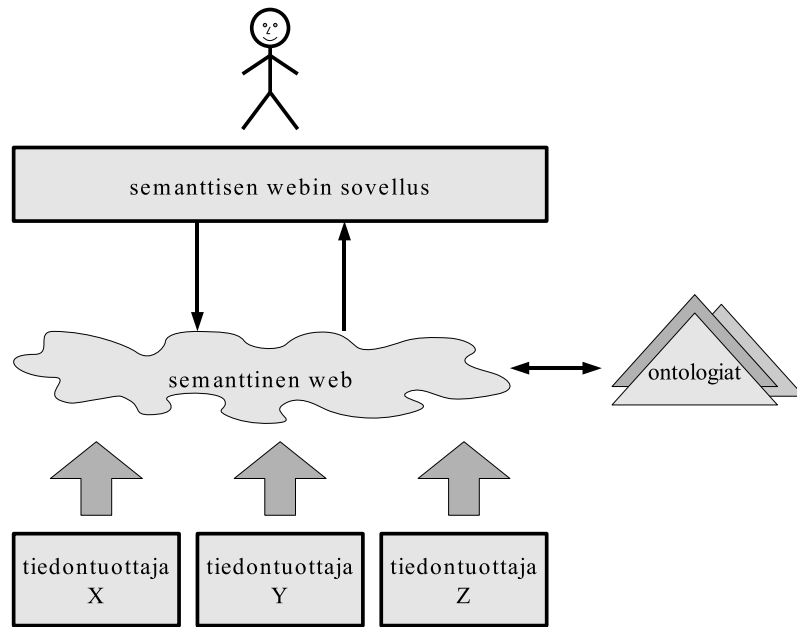
Toinen luottamukseen liittyvä tekniikka on luottamusverkot, jotka kuvaavat henkilöiden välistä luottamusta. Jos esimerkiksi henkilö A luottaa henkilöön B ja B puolestaan henkilöön C, voidaan tämän tiedon avulla päätellä transitiivisesti, että henkilöiden A ja C välillä on todennäköisesti myös ainakin jonkinasteinen luottamus [GKRT04].

4.3 Semanttisen webin erityispiirteitä

Kun eri tiedontuottajat hyödyntävät edellisessä luvussa kuvattuja tekniikoita tiedon esittämiseen, on tuloksena semanttisesti yhteenkytkeytyvä verkkomuotoinen tietokokoelma samaan tapaan kuin nykyinen WWW muodostaa dokumenttien välisen verkon (kuva 7). Koska semanttisen webin tieto tuotetaan hajautetusti, liittyy semanttiseen webiin muutamia erityisongelmia, joita esimerkiksi keskitetyssä tietokantasovelluksessa ei ole.

Yksi nykyisen webin suuria innovaatioita oli se, että rikkinäiset linkit sallittiin, jolloin ei tarvita keskitettyä linkkihallintaa takaamaan linkkien maailmanlaajuista eheyttä. Sama epäeheyden sallimisen idea on myös semanttisessa webissä, jossa URI:en ei tarvitse olla olemassa verkko-osoitteina (URL) ja kaikkea tiettyyn URI:in liittyvää tietoa ei välttämättä ole mahdollista saada selville. Lisäksi eri lähteistä kerätyt tiedot voivat olla keskenään ristiriitaisia esimerkiksi siten, että samalla URI:lla tarkoitetaan eri järjestelmissä eri asioita.

Mikäli eri järjestelmät käyttävät eri ontologioita kuvaamaan samoja asioita voidaan



Kuva 7: Semanttisen webin sovellus hyödyntää alla olevaa virtuaalista maailmanlaajuista, verkottunutta tietosisältöä.

tilanne ratkaista ontologiakytkennöillä, jossa yhteneväiset käsitteet kytketään semanttisesti toisiinsa. On esitetty, että tulevaisuudessa semanttisessa webissä olisi käytettävissä eräänlaisia ontologisia käännöspalveluita, jotka kääntävät yhden ontologian mukaisesti esitetyt tiedot toisen ontologian mukaiseen muotoon [Fen01]. Eli jos semanttisen webin järjestelmä tarvitsisi tiedon siitä, mikä on jonkin tietyn käsitteen URI toisessa ontologiassa, voisi käännöspalvelulta pyytää tätä tietoa.

Tietämyskannoissa ja niihin liittyvissä päättelykoneissa käytetään yleensä suljetun maailman olettamusta (closed-world assumption), joka tarkoittaa sitä, että jos jotain väitettä ei voi todistaa todeksi, se oletetaan epätodeksi. Semanttisessa webissä ei ole järkevää tehdä suljetun maailman oletusta, koska kaikkea webissä olevaa tietoa ei voida koskaan selvittää. Tästä johtuen semanttisessa webissä käytetään yleisellä tasolla avoimen maailman olettamusta (open-world assumption): jos väitettä ei voida osoittaa todeksi, väite ei kuitenkaan välttämättä ole epätosi.

Toinen ero perinteisten sovellusten ja semanttisen webin sovellusten välillä on se, että semanttisessa webissä ei tehdä uniikin tunnisteiden oletusta (unique names assumption). Tästä seuraa, että samaan resurssiin voidaan viitata useilla eri URI:illa. Ongelman hallitsemiseksi on esimerkiksi OWL-kielessä mahdollista ilmaista `owl:sameAs`-predikaatilla, että kaksi URI:a viittaavat samaan käsitteeseen [OWL]. Tätä tietoa hyödyntämällä päättelykoneet voivat toimia oikein myös silloin, kun samalla resursilla on useita URI-tunnisteita.

Käytännössä yksittäisessä sovelluksessa voi olla järkevää käyttää suljetun maailman ja uniikkien tunnisteiden olettamusta, jolloin voidaan hyödyntää perinteisiä päättelykoneita, kuten esimerkiksi Prologia. Tämä oletus tehtiin esimerkiksi MuseoSuomi-järjestelmässä, jota käsitellään luvussa 5.

4.4 Suosittelevat järjestelmät semanttisessa webissä

Semanttisessa webissä toimivat suosittelujärjestelmät ovat järjestelmiä, jotka hyödyntävät suositusten tuotantoon semanttisessa webissä hajautetusti sijaitsevaa tietoa. Yhteistoiminnallisen suosittelun tapauksessa semanttisen webin tekniikoilla esitetään tietoa käyttäjäprofileista eli käyttäjien tekemistä arvioista kohteista [Zie04]. Luonnollisesti myös arvioitavat kohteet voivat sijaita hajautetusti semanttisessa webissä.

Semanttisen webin arkkitehtuurin ja toimintaperiaatteiden takia suosittelujärjestelmien toteutukseen liittyy useita haasteita, joita perinteisissä keskitettyyn tietoon perustuvissa järjestelmissä ei ole.

Semanttisessa webissä tiedon määrä voi olla huomattavasti suurempi kuin perinteisissä suosittelujärjestelmissä. Esimerkiksi yhteistoiminnallisessa suosittelussa hyödynnettäviä käyttäjäprofileja voi olla miljoonia tai tulevaisuudessa jopa miljardeja, mikäli jokainen Internet-käyttäjä julkaisisi profiilinsa (esimerkiksi anonymisoituna).

Perinteisissä yhteistoiminnallisen suosittelun algoritmeissa naapurustot luodaan siten, että jokaista profilia verrataan muihin profileihin samanhenkisten käyttäjien tunnistamiseksi. Isoilla tietomäärillä tämä ratkaisu ei skaalaudu tarpeeksi [ZG05]. Ziegler ja Golbeck esittävät ratkaisuksi profiilien esikarsintaa luottamuksen perusteella siten, että vain luotettavaksi arvioitujen henkilöiden profileja hyödynnetään naapurustoa luotaessa. He osoittavat tutkimuksessaan, että luottamus henkilöiden välillä yleensä heijastaa myös yhteistä makua, joten luottamus toimii mielekkäänä esikarsintaperusteena. Luottamus henkilöiden välillä voidaan mallintaa esimerkiksi siten, että henkilöt kertovat eksplisiittisesti tai implisiittisesti, kehen he luottavat. Luottamusverkkoa seuraamalla voidaan tämän jälkeen löytää kohtuullisen rajallinen määrä luotettavia henkilöitä, jotka ovat potentiaalisesti myös samanhenkisen naapuruston jäseniä. Luottamukseen perustuvan esisuodatuksen jälkeen potentiaalisten henkilöiden määrä on pienempi, jolloin suositteluun voidaan käyttää esimerkiksi perinteisiä yhteistoiminnallisen suosittelun menetelmiä.

Toinen ongelma semanttisen webin laajuudessa toimivissa suosittelujärjestelmissä on se, että käyttäjäprofiilit ovat väistämättä harvoja eli niissä on vain vähän yhteisiä arvioituja kohteita [ZG05]. Tämä johtuu siitä, että potentiaalisia arvioitavia kohteita on miljardeja, mikäli oletuksena on se, että käyttäjä voi arvioida minkä tahansa kohteen, johon voi osoittaa URI:lla. Pelkästään WWW-sivuja on miljardeja ja tämän lisäksi URI:lla voidaan viitata myös kirjoihin, elokuvaan, kulutustuotteisiin jne. Koska profiilit ovat harvoja, on samanhenkisten käyttäjien löytäminen haastavaa. Ziegler ja Golbeck esittävät ratkaisuksi profiilien yleistämistä ontologioiden avulla siten, että yksittäisten kohteiden sijaan profileissa kuvattaisiin, kuinka hyvin mikäkin ontologinen käsite vastaa käyttäjän mielenkiintoa. (Idea on sama kuin luvussa 2.5 mainittu käyttäjäprofiilien yleistäminen demografisten ryhmien avulla.) Lisäksi yksittäisen ontologisen käsitteen käyttäjältä saama arvio propagoidaan yläkäsitteisiin, jolloin profiilin peittävyys suhteessa muiden käyttäjien profileihin

kasvaa. Esimerkiksi kaksi henkilöä, jotka arvioivat samaan *sci-fi*-genreen kuuluvat elokuvat hyväksi, ovat todennäköisesti kiinnostuneita yleisemminkin kyseisestä genrestä. Propagoitumisen ansiosta *sci-fin* yläkäsitteiden, esimerkiksi *elokuvat*, paino profiilissa kasvaa myös.

Edellä kuvattujen erityisesti suosittelujärjestelmiin liittyvien ongelmien lisäksi on myös huomioitava semanttisen webin normaalit haasteet, kuten eri lähteistä kerätyn tiedon ontologisen yhteensopivuuden ja tiedon luotettavuuden varmistaminen, joita käsiteltiin edellä.

Semanttista webiä hyödyntäviä suosittelujärjestelmiä on toteutettu toistaiseksi vain harvoja. Ziegler ja Golbeck ovat tutkineet yhteistoiminnalliseen suositteluun liittyviä ongelmia, kun käyttäjäprofiilien tiedot sijaitsevat hajautetusti semanttisessa webissä [ZG05]. Edellä kuvattiin tärkeimmät ongelmat ja niihin liittyvät ratkaisut heidän tutkimuksessaan.

Ghita et al. ovat puolestaan tutkineet, miten suosittelun kohteisiin liittyvää rikasta kontekstitietoa voitaisiin esittää semanttisen webin välinein ja välittää tietoa sosiaalisissa verkostoissa siten, että tähän tietoon perustuen voitaisiin tehdä suosituksia käyttäjän tietokoneella sijaitsevista dokumenteista [GNP05]. Heidän järjestelmänsä ideana on se, että käyttäjän tietokoneella oleviin dokumentteihin liitetään metatietoa, joka kuvaa dokumentin kontekstin. Esimerkiksi tieteellisten artikkeleiden osalta kontekstiin kuvataan mm. tiedot siitä, mikä dokumentin otsikko on, ketkä ovat sen kirjoittajia, mihin muihin teoksiin artikkelissa viitataan, missä artikkeli on alunperin julkaistu sekä miten dokumentti on hankittu (esimerkiksi WWW-osoite, josta dokumentti on haettu). Kontekstitiedon avulla voidaan päätellä, miten dokumentit liittyvät toisiinsa, jolloin käyttäjälle voidaan suositella esimerkiksi niitä artikkeleita, jotka on julkaistu käyttäjän kannalta tärkeissä konferensseissa.

Varsinainen hyöty järjestelmästä syntyy siinä vaiheessa, kun esimerkiksi saman tut-

kimusryhmän toinen jäsen lähettää hyväksi havaitsemansa artikkelin ja siihen liittyvän kontekstitiedon esimerkkikäyttäjällemme. Artikkelin kontekstitietoja hyödyntämällä sekä lähettäjän yleisen luotettavuuden perusteella lasketaan kuinka suositeltava kyseinen artikkeli on. Kun artikkeleita kerääntyy paljon ja niiden lähettäjiä on useita, voidaan järjestelmällä hallita aiempaa paremmin käyttäjän tietokoneelle muodostunutta “henkilökohtaista digitaalista kirjastoa”, kuten Ghita et al. tilannetta kuvaavat.

Ghita et al. hyödyntävät suositusten tekemisessä kontekstitietojen ja aihealueen ontologioiden semanttisia suhteita. Lisäksi he määrittelevät kullekin ontologiselle suhteelle painokertoimen, jolloin tätä tietoa voidaan hyödyntää suositusten järjestämisessä. Myös käyttäjille voidaan määritellä luotettavuuskerroin, jolla ilmaistaan miten paljon kunkin käyttäjän suosituksiin luotetaan. Järjestelmä on hybridi yhteistoiminnallisen suosittelun ja tietämuspohjaisen suosittelun välillä, jossa suosittelun pohjana käytetään semanttisen webin tekniikoilla esitettävää tietoa.

Dolog et al. ovat puolestaan kehittäneet logiikkapohjaisen hypertekstijärjestelmän, joka tuottaa hypertekstilinkkejä semanttisen webin tekniikoilla esitetystä sisällöstä [DHN03]. He mallintavat ontologioilla dokumenttien sisällön, käyttäjän kontekstitietoja ja järjestelmän tuottamat hypertekstirakenteet. Järjestelmä tuottaa hypertekstirakenteita loogisten sääntöjen avulla, jotka hyödyntävät sekä käyttäjän tilaa että tietoa dokumenteista. Hypertekstirakenteet voivat olla erilaisia linkkejä dokumenttien välillä.

Esimerkkinä tutkimuksessa käytetään koulutuskäyttöä, jossa käytettävissä on koelma dokumentteja kuten opetusaineistoa, harjoitustehtäviä ja koetehtäviä. Opetusaineistoon liittyvät käsitteet, kuten *harjoitustehtävä*, on mallinnettu ontologiaan. Tätä tietoa hyödyntämällä järjestelmä tuottaa linkkejä dokumenteista toiseen huomioiden myös käyttäjän kontekstin siten, että esimerkiksi käyttäjän taustatiedot vaikuttavat käyttäjälle näytettäviin linkkeihin.

5 MuseoSuomen suosittelujärjestelmä

Semanttinen web muodostaa maailmanlaajuisen *tietämyskannan*, joten luvussa 2.7 kuvattu tietämyspohjainen suosittelu sopii luontevasti myös semanttiseen webiin. Tätä ideaa hyödynnetään MuseoSuomi⁸-portaalin suosittelujärjestelmässä.

5.1 MuseoSuomi yleisesti

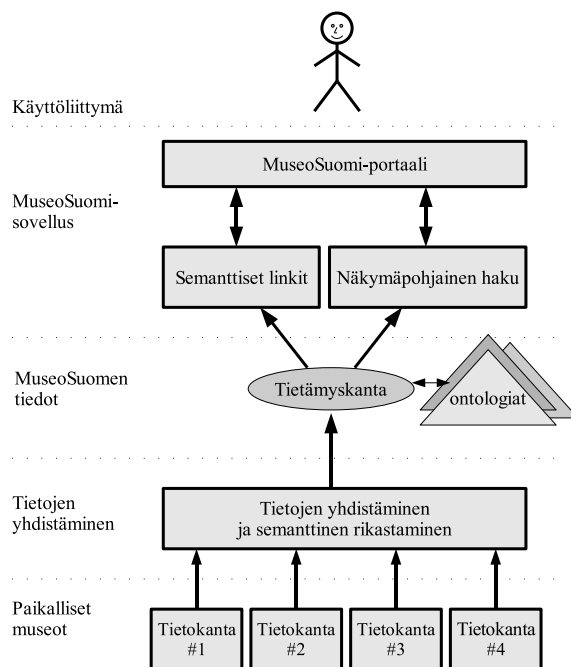
MuseoSuomen ideana on yhdistää useiden autonomisten museoiden kokoelmat saumattomasti yhdistetyksi virtuaaliseksi web-näyttelyksi, jossa voi katsoa ja hakea museoalan kokoelmia sekä samoilla kohteiden välillä [HMS⁺05]. Järjestelmä hyödyntää semanttisen webin tekniikoita kuten RDF:ää ja ontologioita, joiden avulla kunkin mukana olevan museon kokoelmatietokannat on yhdistetty semanttisesti toisiinsa. Käyttäjälle järjestelmä tarjoaa helpon ja yhtenäisen tavan hakea ja tutustua museokokoelmiin riippumatta siitä, mistä kokoelmasta kukin esine on lähtöisin.

MuseoSuomen demojärjestelmä sisältää tällä hetkellä 4000 museoesinettä, jotka ovat esimerkiksi tekstiilejä, huonekaluja ja työkaluja. Lisäksi mukana on 256 muinaismuistokohdetta, jotka ovat esimerkiksi linnavuoria ja hautaröykkiöitä. Aineistot ovat peräisin Museovirastosta, Espoon kaupunginmuseosta ja Lahden kaupunginmuseosta.

MuseoSuomen esine- ja muinaismuistokohteet ovat peräisin neljästä eri tietokannasta, jotka ovat osittain eri organisaatioiden hallinnassa ja joissa käytetään eri luettelointijärjestelmiä, eri tietokantajärjestelmiä ja -malleja. Tulevaisuudessa järjestelmään on tarkoitus kytkeä myös muita kulttuurisisältöjä, kuten Kalevalaa ja Valtion taidemuseon taidekokoelmia.

MuseoSuomi-järjestelmä (kts. kuva 8) koostuu kolmesta pääkomponentista, jotka ovat 1) eri tietokantojen tietojen yhdistäminen ja semanttinen rikastaminen, 2)

⁸<http://www.museosuomi.fi>



Kuva 8: MuseoSuomen arkkitehtuuri

MuseoSuomen tietämyskanta ja ontologiat sekä 3) MuseoSuomi-portaalisovellus tietojen esittämiseen loppukäyttäjälle.

Tietojen yhdistäminen ja semanttinen rikastaminen perustuu MuseoSuomessa kunkin esineen rakenteisiin tekstikuvauksiin, joiden perusteella esine liitetään MuseoSuomen ontologioissa määriteltyihin käsitteisiin. Taulukko 4 antaa mielikuvan esineisiin liittyvistä tiedoista. Esineen semanttinen rikastaminen perustuu siihen, että ontologiassa olevat käsitteet ovat suhteessa muihin käsitteisiin esimerkiksi aliluokka- ja osakokonaisuussuhteiden kautta. Esimerkiksi kun jouluun liittyvä esine liitetään ontologiseen *joulu*-käsitteeseen, joka vuorostaan liittyy *juhliin*, voidaan MuseoSuomi-portaalissa päätellä, että kyseinen jouluesine liittyy myös *juhliin*. Osa esinekorttien kytkennöistä, kuten käyttötilanne, ei perustu suoraan alkuperäisiin museotietoihin. Esimerkkiesineen kytkentä käyttötilanteeseen *joulu* perustuu siihen, että kyseisellä esineellä on määriteltynä asiasana *joulukoristeet*, joka puolestaan on kytketty museoalan asiantuntijan toimesta ontologiassa esiintyvään käyttötilannekäsitteeseen

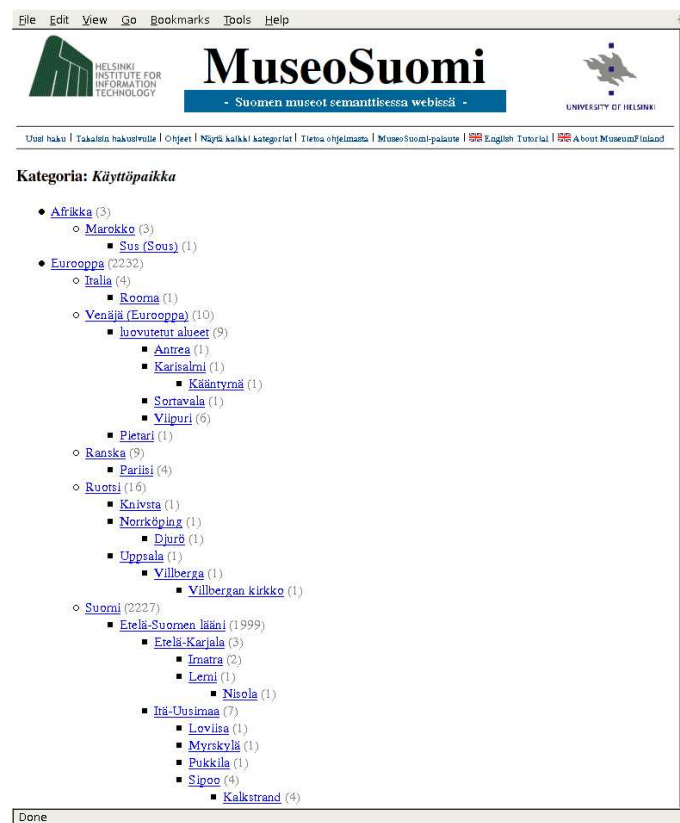
| ominaisuus | ontologia | esimerkki |
|-----------------|------------|---|
| esineytyyppi | Esine | joulukoristeet |
| materiaali | Materiaali | kartonki |
| valmistaja | Toimija | Suomen Korttipalvelu Oy; Heljä Liukko-Sundström |
| valmistusaika | Aika | 1990-luku; uusi aika |
| valmistuspaikka | Paikka | Ulvila |
| käyttäjä | Toimija | Tuulikki Kiveliö; Joonas Nisonen |
| käyttöpaikka | Paikka | Lahti |
| käyttötilanne | Tilanne | joulu; ajanmittaus |
| kokoelma | Kokoelma | Lahden historiallisen museon kokoelmat |

Taulukko 4: MuseoSuomeen tallennetun joulukalenterin semanttiset kytkennät.

joulu.

MuseoSuomen tietämuskanta sisältää kaikki järjestelmään kuuluvat, eri tietokannoista kerätyt esinetiedot sekä näiden semanttiset kytkennät MuseoSuomen ontologioihin. Taulukossa 4 on lueteltuna esineiden ominaisuudet ja niihin liittyvät ontologiat. Lisäksi taulukossa on kunkin ominaisuuden kohdalla esimerkki ontologian käsitteistä, joihin esimerkkiesine on kytketty. Tietämuskanta on esitetty RDF-muodossa ja ontologiat RDFS-muodossa. Ontologioissa on yhteensä yli kymmenen tuhatta luokkaa ja instanssia.

Portaalisovellus koostuu kolmesta päätoiminnosta: semanttisesta OntoViews-portaalikoneesta [MHSV04], Ontogator-näkymäpohjaisesta hakukoneesta sekä Ontodella-päätelykoneesta. Näistä OntoViews huolehtii käyttöliittymän esittämisestä ja käyttäjän interaktioiden käsittelystä. Ontogator puolestaan suorittaa käyttäjän antamien hakuoptioiden perusteella ns. leikkauksia alla olevaan dokumenttiaineistoon palauttaen ainoastaan ne dokumentit, joissa kaikki annetut hakuehdot ovat voimas-



Kuva 9: MuseoSuomen “Käyttöpaikka”-näkymä

sa. Ontodellaa käytetään näkymien generointiin sekä semanttisten linkkien generointiin eli tietämyspohjaiseen suositteluun.

Näkymien generointi tarkoittaa sitä, että ontologioiden ja museoaineiston muodostamasta verkosta tuotetaan loogisten sääntöjen avulla käyttäjälle näytettäviä toisiaan leikkaavia näkymiä, joiden avulla esineitä voi hakea. Esimerkiksi käyttöpaikkanäkymä (kuva 9) sisältää paikkahierarkian, jossa ylimpänä ovat maanosat, jonka alla ovat maat, läänit, maakunnat, kunnat, kaupungit, kylät, kaupunginosat jne hierarkisessa suhteessa toisiinsa. Näkymä perustuu luvussa 4.2.3 kuvattuun MuseoSuomen paikkaontologiaan. Kuhunkin hierarkian solmuun on kytketty kaikki ne kohteet, jotka liittyvät loogisen säännön kuvaamalla tavalla kyseiseen paikkaan.

Loppukäyttäjälle MuseoSuomi tarjoaa tehokkaita haku- ja selailutoimintoja sekä



Kuva 10: MuseoSuomi, etusivu

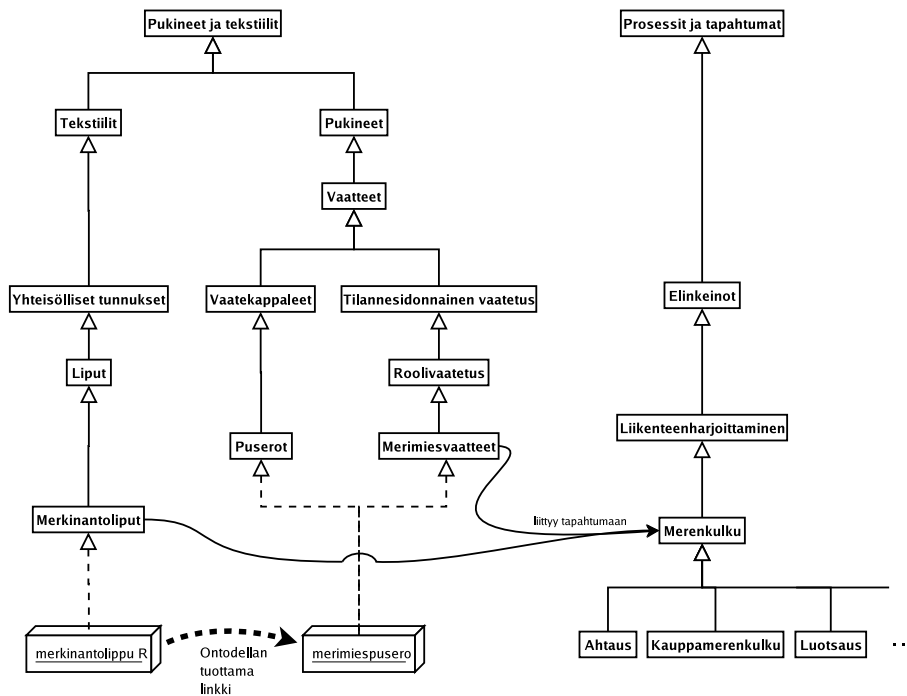
WWW- että WAP-selaimella käytettäväksi. Kuvissa 10 ja 11 näkyy MuseoSuomen WWW-hakukäyttöliittymä, jonka avulla käyttäjä voi tehdä näkymähierarkioita avaamalla ja käsitteitä hierarkioista valitsemalla ns. näkymäpohjaisia hakuja alla olevaan aineistoon. Käyttäjä voi tehdä valintoja useista näkymistä samaan aikaan, jolloin hakutulokseen tulevat ainoastaan ne kohteet, jotka täsmäevät kaikkiin hakuehtoihin. Esimerkiksi kuvassa 11 käyttäjä on valinnut haluavansa nähdä työvälineet. Tämän jälkeen hän voisi rajata haluavansa nähdä ainoastaan työvälineitä esimerkiksi tietyltä aikakaudelta tai tiettyyn paikkaan liittyen.

MuseoSuomen esinesivu (kuva 12) sisältää esineen kuvan ja kuvaustekstit, tiedot hakunäkymistä, joihin esine kuuluu, sekä tietämuspohjaisesti tuotetut semanttiset linkit muihin esineisiin (kuvassa oikealla). Semanttisten linkkien avulla käyttäjä voi liikkua horisontaalisesti kokoelman sisällä saaden hyvän mielikuvan siitä, miten esineet kytkeytyvät monimuotoisesti toisiinsa.



Kuva 11: MuseoSuomi, tulossivu

Kuva 12: MuseoSuomen esinesivu. Kuvan oikeassa laidassa näkyvät semanttiset lin-
kit, joista alimpana “merimiespusero”-suositus.



Kuva 13: Liittyy tapahtumaan -säännön avulla tuotetaan linkki kohteesta “Merkinantolippu R” kohteeseen “Merimiespusero”.

5.2 Tietämyspohjainen suosittelu MuseoSuomessa

MuseoSuomen tietämyspohjaisesta suosittelusta vastaa Ontodella-päättelykone⁹, jossa suositukset tuotetaan loogisten sääntöjen ja kohteiden semanttisten ominaisuuksien perusteella. Loogiset säännöt kuvaavat ontologioiden käsitteitä hyödyntäen ehdot sille, milloin kaksi kohdetta linkitetään toisiinsa.

MuseoSuomen suosittelusäännöt esitetään Prolog-predikaatteina¹⁰ (katso luku 4.2.4), jotka voivat olla mielivaltaisen monimutkaisia. Säännöissä hyödynnetään kohteiden metatietoja sekä MuseoSuomen ontologioita. Sääntöjen kuvaaminen ontologiakäsitteitä hyödyntäen erottaa säännöt aineistosta, jolloin (optimitilanteessa) aineisto

⁹Ontodellan toteutuksesta ovat pääsääntöisesti vastanneet Arttu Valo ja allekirjoittanut. Ontodella toteutettiin “Semantic Web: Intelligent Directories”-projektissa vuosina 2002-2004.

¹⁰Esimerkit ovat SWI-Prologin mukaista syntaksia [SWI].

voidaan vaihtaa muuttamatta sääntöjä.

Prolog-muotoisen suosittelusäännön pää (head) on muotoa:

```
rule_name(Subject, Target, Explanation)
```

jossa *rule_name* on säännölle annettu nimi, *Subject* (subjekti) on kohteen URI, jolle suosituksia tuotetaan, *Target* (kohde) on kohde (URI), jota suositellaan ja *Explanation* sisältää selityksen sille, miksi subjekti linkitetään tiettyyn kohteeseen¹¹. Suosituksen selitys vastaa kysymykseen *miksi* kohdetta suositellaan käyttäjälle. MuseoSuomen tapauksessa tuotetaan sekä ihmiselle tarkoitettu sanallinen selitys että koneelle tarkoitettu selitys-URI. Tyypillisesti MuseoSuomen selitys perustuu yhteiseen yläkäsitteeseen, joka löytyy subjektin ja kohteen väliseltä RDF-polulta, jota sekä sanallinen selitys että selitys-URI kuvaavat. Suosittelevien selittäminen selkokielellä on tärkeää, jotta käyttäjä voisi paremmin arvioida, mistä suosituksista hän on kiinnostunut.

Esimerkiksi käyttäjän katsoessa MuseoSuomessa “Merkinantolippu R”-kohdetta, voidaan ontologisen tietämyksen perusteella päätellä, että samaan aiheeseen, merenkulkuun, liittyy myös esine “merimiespusero”. Tämän seurauksena “Merkinantolippu R”-esineen yhteydessä näytetään linkki merimiespuseroon. Kuvassa 13 on graafisesti esitetty miten esineet linkittyvät toisiinsa. Linkitys perustuu seuraavaan päättelyketjuun:

1. *Merkinantolippu R* -esine on *Merkinantolippu*-luokan ilmentymä.
2. *Merkinantolippu*-luokka liittyy *Merenkulku*-luokkaan.
3. Myös *Merimiesvaatteet*-luokka liittyy *Merenkulku*-luokkaan.
4. *Merimiespusero*-esine on *Merimiesvaatteet*-luokan ilmentymä.

¹¹Linkkipredikaatit vastaavat [HVVH04]:ssa kuvailtuja toimintaperiaatteita.

Päätelyn seurauksena *Merkinantolippu R* -esine linkitetään *Merimiespusero*-esineeseen. Linkin tyyppi on “Liittyy tapahtumaan” ja sitä kuvaa *Merenkulku*-luokka, joka oli yhdistävänä tekijänä edellisessä päätelyketjussa. Näin ollen selitystekstinä on “Merenkulku”. Päätelyyn käytettävä sääntö on Prolog-muodossa liitteessä 1.

Ontodellan nykyisessä versiossa suositukset palautetaan ilman mitään relevanssiarvoa siinä järjestyksessä, jossa ne Prologin suorituksen aikana syntyvät. Käytännössä järjestys riippuu siitä, missä järjestyksessä RDF-aineisto on ladattu Ontodellaan. MuseoSuomen käyttöliittymässä näytetään ainoastaan kunkin suositteleyryhmän ensimmäiset viisi linkkiä.

MuseoSuomeen toteutettiin mm. seuraavat suositteley säännöt:

Esineeseen liittyvään paikkaan liittyviä muinaismuistoja Linkki esineeseen liittyvän paikan ja samassa paikassa sijaitsevan muinaismuistokohteen välillä.

Liittyy esinetyyppiin Esineellä on sama esinetyyppi (esimerkiksi “maalaukset”) kuin linkitettävällä esineellä.

Liittyy tapahtumaan Esine liittyy samaan tapahtumaan tai prosessiin (esimerkiksi “häät”, “kirkolliset tapahtumat”, “merenkulku”).

Samaan aiheeseen liittyviä esineitä Tämä linkitys perustuu siihen, että kahdella esineellä on yksi tai useampi sama Museoalan ontologian käsite asiasanat ominaisuutena.

Sama käyttäjä Esineellä on sama käyttäjä kuin linkin kohteena olevalla esineellä.

Sama käyttöpaikka Esineellä on sama käyttöpaikka kuin linkin kohteena olevalla esineellä.

Sama materiaali Esine on valmistettu samasta materiaalista kuin linkin kohteena oleva esine.

Samaan paikkaan liittyviä esineitä Esineet, joilla on sama käyttö- tai valmistuspaikka.

Samaan paikkaan liittyviä muinaismuistoja MuseoSuomi-aineistoon on kytketty testimielessä myös Museoviraston muinaismuistorekisteri. Linkin kohteena ovat tällaiset muinaismuistot.

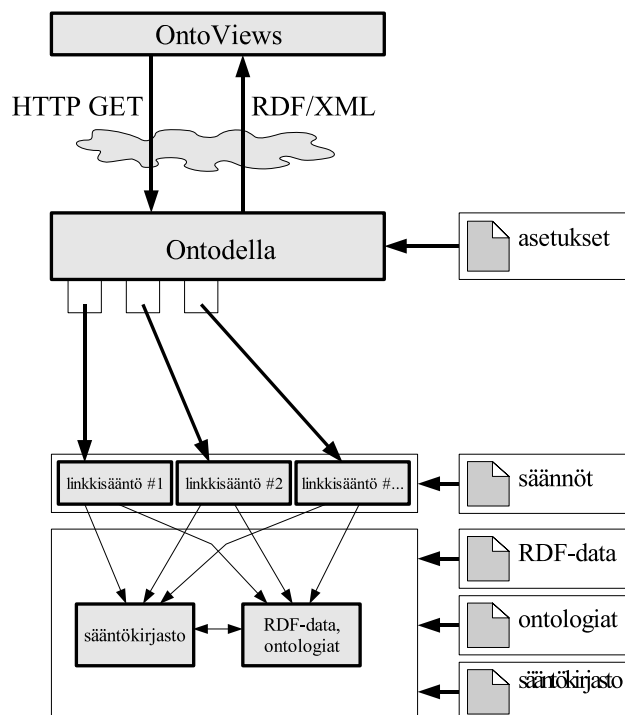
Sama valmistaja Kahdella esineellä on sama valmistaja.

Sama valmistusaika Kahdella esineellä on sama valmistusaika.

Sama valmistuspaikka Kahdella esineellä on sama valmistuspaikka.

Ontodella-järjestelmä on toteutettu teknisesti siten (kts. kuva 14), että Ontodella vastaanottaa HTTP-pyyntöjä, joiden parametrina on sen esineen URI, jolle semanttisia suosituksia halutaan tuottaa. Kyseisen esineen suositukset tuotetaan suorittamalla kaikki Ontodellaan rekisteröidyt linkkisäännöt siten, että subjektiksi asetetaan kyseisen esineen URI. Tuloksena syntyy lista suosituksista, jotka on ryhmitelty ensisijaisesti suosittelusäännön mukaan ja toissijaisesti perustelun mukaan (kuva 12, oikea laita). Lopuksi tulokset serialisoidaan RDF/XML-muotoon (liite 2) siten, että ne voidaan muuntaa suoraviivaisesti HTML-muotoon OntoViewsissä.

Ontodellan säännöt hyödyntävät Ontodellaan ladattuja faktoja (RDF-dataa ja ontologiat) sekä Ontodellan sääntökirjaston sääntöjä, joihin on kerätty yleiskäyttöisempiä sääntöjä, joita tarvitaan useissa eri linkkisäännössä. Tällaisia sääntöjä ovat esim *isArtifact(X)*, joka evaluoituu todeksi mikäli URI *X* on tyypiltään esine sekä *rdfs_transitive_subClassOf(X, Y)*, joka evaluoituu todeksi, jos luokkien *X* ja *Y* välillä on aliluokkasuhde.



Kuva 14: Ontodellan arkkitehtuuri

Ontodellan tehokkuus aika- ja muistivaativuuden näkökulmasta on ollut hyvän ilman erityisiä optimointitoimenpiteitä. Hitain vaihe suorituksessa on RDF-aineiston lukeminen muistiin Ontodellan käynnistysvaiheessa, johon kuluu noin 20 sekuntia aineiston ollessa kooltaan noin 200 000 RDF-kolmikkoo¹². Tyypillinen vastausaika suosittelukyselyihin on noin 1-2 sekuntia. Suosittelesäännöt on pyritty kirjoittamaan siten, että ne voidaan suorittaa tehokkaasti. Tämä tarkoittaa käytännössä Prolog-suorituksen turhien haarojen katkaisuja ja sääntöjen järjestyksen huolellista harkintaa, jotta peräytystä (backtracking) tapahtuisi mahdollisimman vähän. Keskusmuistia SWI-Prolog tarvitsee Ontodellan suorittamiseen noin 50 megatavua.

Ontodellaa on MuseoSuomen lisäksi hyödynnetty muissa OntoViews-pohjaisissa palveluissa, kuten julkisten palveluiden hakuun tarkoitetussa semanttisessa Suomi.fi-

¹²Testauslaitteisto: Pentium IV 2.66 GHz prosessori, jossa 1 Gt keskusmuistia ja käyttöjärjestelmänä Linux. SWI-Prologin versio: 5.4.7.

portaalisovelluksessa [Sid05] sekä Orava-elokuvahakuportaalissa [Ora]. Ontodellan aiempaa versiota on hyödynnetty mm. promootiokuvien katseluun tarkoitettussa kioskisovelluksessa Promoottorissa [HSV04] sekä “köyhän miehen semanttisen webin HTML-generaattorissa” [HVVH04].

5.3 MuseoSuomen suosittelujärjestelmän arviointi

MuseoSuomi-järjestelmä demonstroi yhden tavan toteuttaa useaan tietolähteeseen, semanttisen webin tekniikoihin perustuvan museokokoelmien haku- ja selailupalvelun. MuseoSuomen semanttiset linkit toimivat sääntöpohjaisesti ontologioiden käsitteitä hyödyntäen siten, että alla oleva aineisto on vaihdettavissa, kunhan kohdet kuvataan samojen ontologioiden avulla. Ratkaisu on osoittautunut joustavaksi ja toimivaksi ainakin MuseoSuomi-järjestelmän puitteissa sekä niissä MuseoSuomen sisärjestelmissä, joissa Ontodellaa on hyödynnetty (esimerkiksi semanttinen Suomi.fi-portaali [Sid05]).

Ontologiaan ja sääntöihin perustuvan suosittelun etu on se, että aineistoa ei tarvitse linkittää yksittäisen kohteen tasolla vaan riittää, että kuvataan yleiset säännöt sille, milloin kaksi kohdetta linkittyy toisiinsa. Sääntöpohjainen linkittäminen on mielekästä, mikäli aineisto on tarpeeksi yhdenmukainen (hyödynnetään esimerkiksi yhteisiä ontologioita) ja ontologiat ovat tarpeeksi rikkaita, jotta aineiston mielekäs kuvaaminen on mahdollista (kaikki kohteet eivät esimerkiksi liity samaan ontologiseen käsitteeseen).

Mitä isompi aineisto, sitä suurempi potentiaalinen hyöty ontologiapohjaisesta, käsitetasolla tapahtuvasta semanttisesta linkittämisestä ja suosittelusta voidaan saavuttaa. Esimerkiksi MuseoSuomen noin 4000 kohteen linkittäminen käsin olisi iso urakka, koska jokaisen esineen kohdalla on periaatteessa 4000 potentiaalista linkityksen kohdetta. Lisäksi, koska MuseoSuomen aineisto koostuu eri lähteistä kerätystä ai-

neistosta, ei yksittäisen museon asiantuntija välttämättä tuntisi muiden museoiden aineistoa yhtä hyvin kuin omaansa, jolloin hän saattaisi linkittää oman museon esineitä pääasiallisesti oman museon esineisiin. Sääntöpohjaisen linkittämisen ansiosta eri lähteistä kerätty tieto linkittyy yhtenäiseksi WWW-näyttelyksi.

Aineiston kuvaaminen ontologioilla voi olla työlästä. MuseoSuomen tapauksessa tämä pystyttiin automatisoimaan osittain hyödyntäen esimerkiksi museoalan vakiintuneita asiasanakäytäntöjä. Jos automatisointi ei ole mahdollista, voi aineistojen kuvaaminen ontologioilla olla tästä huolimatta mielekästä, mikäli halutaan hyödyntää esimerkiksi MuseoSuomen demonstroimia semanttisia hakutoiminnallisuuksia ja automaattista, sääntöpohjaista linkitystä.

Ontodellan linkkien tuotannossa haasteeksi osoittautui päättää, mitä linkejä käyttäjälle tulisi tuottaa. Tavoitteena oli tuottaa ”kiinnostavia” (tai jopa ”hämmäntäviä”) linkejä käyttäjän kulloiseenkin esineeseen liittyen, mutta koska kiinnostavuus (hämmäntävyys) on riippuvainen käyttäjästä ja käyttötilanteesta, vaatii se sovellus- alakohtaisia päätöksiä. MuseoSuomi-projektin puitteissa määriteltiin sääntöjä lähinnä siinä määrin, että pystyttiin testaamaan semanttisten linkkien tekninen toiminta. Todellisessa tilanteessa säännöt tulisi määritellä aihealueen asiantuntijan toimesta eli esimerkiksi MuseoSuomen tapauksessa museopedagogista osaamista hyödyntäen. Sääntöjen esittäminen Prologilla oli suoraviivaista ainakin tietojenkäsittelytieteen opiskelijalle. Sääntöjen tuottaminen museoalan asiantuntijan toimesta vaatisi todennäköisesti Prologin peittämistä helppokäyttöisellä käyttöliittymällä, jossa sääntöjen ilmaisuvoimaisuutta todennäköisesti jouduttaisiin vähentämään helppokäyttöisyyden lisäämiseksi. MuseoSuomen osalta Ontodellan säännöissä käytettiin käytännössä varsin rajallisesti Prologin ilmaisuvoimaa. Yleisemmin säännöissä seurattiin joko suoria tai transitiivisia suhteita RDF-verkossa tai kutsuttiin Ontodellan sääntökirjaston sääntöjä (kuten *isArtifact(X)*).

Yksi käytännön havainto sääntöihin liittyen oli se, että sääntöjen monimutkaisuutta lisäsi se, mikäli haluttu linkitys poikkesi huomattavasti annotointien ja ontologioiden rakenteesta. Sääntöjen kirjoittamisen kannalta olisi parempi, mikäli ontologioiden mallinnusvaiheessa tiedettäisiin ainakin suuntaa-antavasti, miten tietoa halutaan sääntöjen avulla myöhemmin hyödyntää. Erityisesti Ontodellan edeltäjässä, Promoottorin [HSV04] suosittelujärjestelmässä oli havaittavissa se, että käytetty ontologia oli mallinnettu sinänsä hyvin tarkasti, mutta suosittelusääntöjen kannalta epäkäytännöllisesti, jolloin säännöistä tuli osittain tarpeettoman monimutkaisia.

Toinen käytännön havainto oli se, että sääntöjen testaaminen ja niiden toimivuuden varmistaminen oli virhealtista, koska sääntöjen toimivuuteen vaikuttivat sekä päätelyn pohjana olevat faktat (museoesineiden annotaatiot että ontologiat) että muut säännöt. Virheitä ilmeni molemmissa. Lisäksi toteutuksessa käytetyn Prolog-tulkin (SWI-Prolog) ilmoitukset käyttäjälle olivat usein epäinformatiivisia. Tyypillisessä virhetilanteessa Prolog-tulkki tulosti ainoastaan vastauksen “no”, joka tarkoittaa sitä, että suoritettu Prolog-predikaatti evaluoitui epätodeksi.

Ontodellan vahvuus moniin muihin suosittelujärjestelmiin verrattuna on se, että kuhunkin tuotettuun linkkiin pystytään tuottamaan sekä selkokielen että koneelle tarkoitettu kuvaus (yhdistävän resurssin URI) siitä, miten linkki liittyy nykyiseen kohteeseen. Nykyisessä MuseoSuomi-toteutuksessa hyödynnettiin vain ihmiselle tarkoitettua selitystekstiä, mutta koneelle tarkoitettua selitystä voisi käyttää esimerkiksi monilähteisessä suosittelussa siten, että jokin muu suosittelujärjestelmä hyödyntäisi Ontodellan tuottamia suosituksia jalostamalla niitä eteenpäin. Koneelle tarkoitettua selitystä voisi jatkokehittää puolestaan siten, että URI:n lisäksi selitys sisältäisi kokonaisen RDF-muotoisen esityksen, joka sisältäisi koko RDF-polun suosituksen subjektista kohteeseen. Tätä tietoa voisi käyttää myös suosittelun perustelun visualisointiin käyttöliittymässä.

Ontodellan palauttamia suosituksia ei tällä hetkellä järjestetä mihinkään relevans-

sijärjestykseen, koska sopivaa järjestelytapaa ei projektin aikana keksitty. Yhtenä vaihtoehtona mietittiin hybridijärjestelmän (luku 2.8) toteuttamista siten, että sääntöpohjaista suosittelua olisi tuettu yhteistoiminnallisen suosittelun avulla. Mahdollisena hyötynä olisi ollut se, että sääntöpohjaisesti tuotetut suositukset olisi voitu järjestää käyttäjän yhteistoiminnallisen profiilin mukaiseen paremmuusjärjestykseen. Tätä ei kuitenkaan testattu käytännössä, koska vaikutti epävarmalta, olisiko käyttäjiltä saatu tarpeeksi implisiittistä tai eksplisiittistä arvostelutietoa käyttäjäprofiilien luomiseksi, jotta yhteistoiminnallinen suosittelu olisi toiminut. Käytännön hankaluus käyttäjäpalautteen keräämiseksi oli myös se, että MuseoSuomi-järjestelmän julkinen versio valmistui projektin päättyessä, joten käyttäjäpalautteen kerääminen ja yhteistoiminnallisen suosittelun testaaminen olisi ollut aikataulun puitteissa hankalaa.

Mahdollisen implisiittisen arvostelutiedon keräämisen osalta jäi myös avoimeksi, mitkä olisivat MuseoSuomen tapauksessa hyviä keinoja mitata käyttäjän kiinnostusta tiettyyn kohteeseen liittyen. Miten esimerkiksi tulkitaan se, että käyttäjä menee tietylle esinesivulle tai seuraa esinesivulla olevia semanttisia linkkejä? Museoesineiden etsiminen poikkeaa normaalista etsinnästä siten, että käyttäjällä ei välttämättä ole kovin tarkkaa tavoitetta sille, mitä hän on etsimässä. Tilanne on toinen, kuin jos käyttäjä olisi esimerkiksi MuseoSuomi-tyyppisellä järjestelmällä hakemassa keltaiset sivut -palvelusta autokorjaamo rikkiäisen auton korjaamiseksi, jolloin tiedonhakarve on varsin yksiselitteinen. MuseoSuomi on ennemminkin ajanvietepalvelu, johon tullaan joksikin aikaa viihtymään ja oppimaan entisajan elämästä.

Kuten edellä todettiin, ovat Ontodellan käynnistymis- ja suoritusajat varsin kohtuulliset. Tarvittaessa niitä voisi mahdollisesti lyhentää siten, että aineistosta ja ontologioista suodatettaisiin pois kaikki ne tiedot, joita suosittelusäännöissä ei hyödynnetä. Tällaisia tietoja ovat MuseoSuomen suosittelusääntöjen osalta esimerkiksi RDF-kolmikot, joissa objektin arvona on merkkijono. Mielenkiintoinen ongelma on,

missä määrin suositteluun käytettäviä Prolog-sääntöjä voisi analysoida siten, että aineiston ja ontologioiden suodatin voitaisiin tuottaa automaattisesti.

Ontodellan vasteaikoja suosittelupyyntöihin voidaan tarvittaessa tehostaa siten, että suosittelut kullekin subjektille tallennetaan välimuistiin, jolloin ne ovat saatavissa sieltä nopeasti. Välimuistina voitaneen käyttää normaalia HTTP-välimuistia (kuten Squid [Squ]), koska kyselyt Ontodellalle tehdään HTTP-muotoisina kyselyinä. Tarvittaessa Ontodellan suositukset voidaan myös laskea etukäteen kaikille subjekteille. Näin toimittiin Ontodellan edeltäjässä, Promoottorin suosittelujärjestelmässä [HSV04]. Välimuisteihin ja etukäteen laskemiseen liittyy haasteita silloin, jos suositusten tuotantoon käytettävät tiedot muuttuvat dynaamisesti. Ontodellan nykyinen versio tukee tiedon muuttamista, koska suositukset tuotetaan joka pyynnön yhteydessä uudelleen. MuseoSuomi-projektin puitteissa välimuisteja tai etukäteen laskemista ei Ontodellaan liittyen testattu.

Ontodella on hajautetusti tuotettuun tietoon perustuva keskitetty tietämuspohjainen suosittelujärjestelmä. Luonteva jatkokehitysaskel olisi se, että myös varsinaista suositusten tuotantoa voitaisiin hajauttaa siten, että päättely tapahtuisi hajautetusti. Ontodella toimii jo nyt HTTP-rajapinnan yli eli periaatteessa OntoViews voisi ottaa yhteyttä HTTP:n yli useaan eri Ontodella-järjestelmään ja pyytää kultakin suosituksia annettuun URI:in liittyen. Tällaista useiden Ontodella-instanssien käyttöä ei MuseoSuomi-projektin puitteissa testattu.

Verrattuna luvussa 4.4 mainittuihin yhteistoiminnalliseen suositteluun perustuviin Zieglerin ja Golbeckin sekä Ghitan et al. järjestelmiin on suurin ero se, että Ontodellassa suositukset tuotetaan tietämuspohjaisesti, loogisten sääntöjen avulla. Ghita et al. hyödynsivät suosittelussa myös dokumenttien välisiä semanttisia suhteita mitaten niiden semanttisia etäisyyksiä toisistaan. Ontodellassa semanttista etäisyyttä ei hyödynnetä. Ontodellan kehittämisen aikana pohdittiin semanttisen etäisyyden hyödyntämistä järjestelmässä siten, että tulokset olisi järjestetty etäisyyden perus-

teella. Tästä luovuttiin, koska semanttisen etäisyyden määrittely pelkästään RDF-verkon solmujen välisenä etäisyytenä ei ole mielekäästä, sillä etäisyys kahden solmun välillä voi vaihdella ilman, että verkon semanttinen tulkinta muuttuisi. Esimerkiksi yläluokan X ja aliluokan Y välillä voi olla nolla tai useampia luokkia, mutta kuitenkin aliluokkasuhde $X:n$ ja $Y:n$ välillä pysyy voimassa. Eri ontologioissa voi olla tarpeen kuvata eri tarkkuudella asioita, joten tästä johtuen myös käsitteiden määrä vaihtelee.

Dologin et al. järjestelmä muistuttaa toiminnallisuudeltaan Ontodellaa. Molemmissa käytetään loogisia sääntöjä ja semanttisen webin sisällöillä kuvattua sisältöä linkkien tuottamiseen. Edistyksellisempää Dologin et al. järjestelmässä on se, että käyttäjän konteksti kuvataan tarkemmin kuin Ontodellassa. Ontodellassa käyttäjän tilaa edustaa sen esineen URI, jota käyttäjä sillä hetkellä katsoo ja jolle suosituksia tuotetaan.

Molemmissa järjestelmissä suosituksen tulokset esitetään RDF-rakenteena. Ontodellassa tämä rakenne välitetään käyttöliittymäkerrokselle, joka tuottaa linkeistä HTML-muotoisen esityksen. Dologin et al. järjestelmässä ei kuvailla miten tuotetut hypertekstirakenteet esitetään käyttäjälle.

Ontodellassa aineisto ja ontologiat ovat isompia ja monipuolisempia kuin Dologin et al. järjestelmässä. MuseoSuomen ontologioilla kuvataan pääsääntöisesti aineiston sisältöä, kuten esimerkiksi museoesineen käyttötarkoitusta ja ikää, kun taas Dologin et al. järjestelmässä aineisto mallinnetaan pääasiallisesti käyttötarpeen kannalta, kuten esimerkiksi käsitteellä *harjoitustehtävä*.

Ontodellassa sisältö on lähtöisin useasta eri tietolähteestä. Dologin et al. järjestelmässä aineisto on (ilmeisesti) tuotettu yhden tahon toimesta.

6 Monilähteinen suosittelu semanttisessa webissä

Luvussa 3.2 kuvattiin haasteita, joita monilähteisiin suosittelujärjestelmiin liittyen on havaittu etenkin MetaLens-järjestelmän puitteissa. Ydinongelma on, miten hyödyntää useasta tietolähteestä saatavia tietoja suositusten tuottamiseen.

Semanttisessa webissä ideana on ratkaista usean tietolähteen hyödyntämisen ongelma luomalla nykyisen webin päälle "semanttinen" kerros, jossa tieto kuvataan yhteisten käsitteiden (ontologiat) avulla ja yhteisen kielen (RDF) avulla. Ratkaisu vaikuttaa sovelialta myös monilähteiseen suositteluun, joten seuraavassa tarkastellaan miten luvun 3.2 haasteet voidaan ratkaista semanttisen webin keinoin:

Haaste 1. Miten autonomisia tietolähteitä hyödynnetään suosittelussa?

Semanttisessa webissä autonomisia tietolähteitä on joustavaa hyödyntää, koska tietosisältö esitetään yhteisessä muodossa ja yhteisissä ontologioissa määritellyjä käsitteitä käyttäen. Tällöin vältytään esimerkiksi MetaLens-järjestelmässä (luku 3.1) käytettyjen epävarmojen kääresovellusten käytöltä. Luonnollisesti ongelmia syntyy, mikäli eri järjestelmät käyttävät eri ontologioita tietojen kuvaamiseen, kuten luvussa 4.3 todettiin.

MuseoSuomi-järjestelmä on esimerkki siitä, miten tämä tiedon yhdistäminen voidaan toteuttaa käytännössä ja miten tietämyspohjainen suosittelujärjestelmä voi hyödyntää semanttisen webin tekniikoilla kuvattua tietoa.

Kun tieto on saatu käsitteellisesti yhtenäiseen muotoon, on seuraava kysymys se, onko tieto luotettavaa. Tähän liittyen Ziegler ja Golbeck pohtivat, miten semanttisessa webissä mallinnetaan tiedon luotettavuutta ja miten tätä tietoa hyödynnettäisiin suositusten tuotannossa. MuseoSuomessa tiedon luotettavuuden ongelma sivuutettiin käytännössä siten, että tiedon tuottajia, kuten Museovirastoa ja Espoon kaupunginmuseota, pidettiin lähtökohtaisesti luotettavina, oman aihealueensa

asiantuntijoina. Luonteva jatkokehityskohde MuseoSuomessa olisi se, miten voitaisiin hyödyntää semanttisessa webissä olevia muita tietolähteitä siten, että luotettavuus voitaisiin säilyttää. Kyseeseen voisi tulla esimerkiksi harrastajien, koululaisten ja yhdistysten tuottamat historiaan liittyvät aineistot. Esimerkiksi Kirkkonummella Sepän koulun koululaiset ovat tehneet kiinnostavan WWW-sivuston Porkkalan vuokraukseen vuosina 1944-1956 liittyen [Por]. Voisiko MuseoSuomen esineet linkittää tulevaisuudessa myös tällaiseen sisältöön?

Semanttista webiä hyödyntäen ollaan myös askeleen lähempänä sitä, että käyttäjä itse voisi valita, mitä lähteitä hän haluaa käytettävän suositusten tuottamisessa. Käyttäjä voisi syöttää suosittelujärjestelmälle niiden tietolähteiden URL:it, joita hän haluaa hyödyntää suosittelussa. Esimerkiksi elokuvasuosituksia etsiessään käyttäjä voisi antaa paikallisen elokuvateatterin URL:in suosittelujärjestelmälle. Mikäli elokuvateatteri esittäisi RDF-muodossa, mitä elokuvia teatterissa on tarjolla, voisi suosittelujärjestelmä valita käyttäjän maun huomioiden ehdotuksen siitä, mikä elokuva paikallisen teatterin tarjonnassa vastaa parhaiten käyttäjän makua. Tällaisen avoimen monilähteen semanttisen webin suosittelujärjestelmän tekeminen olisi luonteva jatkoaskel tässä tutkielmassa kuvatuille ajatuksille ja liittyy läheisesti edellä kuvattuun tiedon luotettavuuden ongelmaan.

Haaste 2. Miten eri lähteistä tulevat sisällöt yhdistetään?

Semanttisen webin ideana on kuvata tieto URI:ien muodostaman yhtenäisen verkkomaisen tietorakenteen eli RDF:n avulla. RDF:n perusominaisuus on se, että kaikki RDF-muotoiset tiedot ovat yhdistettävissä toisiinsa, koska kahdessa eri RDF-verkossa olevien samanarvoisten URI:en oletetaan olevan universaalisti identtiset ja viittaavan samaan resurssiin.

Tietojen yhdistäminen useista eri lähteistä voi sisältää, paitsi luotettavuusongel-

mia tiedon oikeellisuuden suhteen, myös juridisia ongelmia esimerkiksi tietosuojan ja tekijänoikeuteen liittyen. Esimerkiksi henkilörekisterin tietoja ei lähtökohtaisesti saa yhdistää toiseen rekisteriin eikä sukulaisuussuhteita päätellä aineistosta, vaikka se ehkä teknisesti olisikin mahdollista. Teknisestä näkökulmasta katsottuna tietojen yhdistämistä on vaikeaa estää sen jälkeen, kun tieto on kerran julkaistu. On siis tarkkaan harkittava, mitä mahdollisesti suoraan tai epäsuorasti arkaluontoisia tietoja julkaistaan ja missä muodossa.

Tekijänoikeuslain osalta kysymykseen voi tulla esimerkiksi sisällön suojaaminen salakirjoituksella siten, että käyttäjä pystyy purkamaan sisällön vasta maksettuaan asianmukaisen korvauksen materiaalista. Esimerkiksi DiMaS-järjestelmässä ideana on luoda videotiedostojen jakoon kaksiosainen jakelupaketti, jossa videon metatiedot ja mainosvideo (traileri) ovat salaamattomana, kaikkien nähtävillä, mutta varsinainen sisältö on salakirjoitettu laittoman käytön estämiseksi [RS04]. Metatiedon ja trailerin avulla käyttäjän on mahdollista löytää haluamansa video muiden videoiden joukosta. Halutessaan hän voi tämän jälkeen ostaa purkuavaimen tekijältä, jonka jälkeen hän voi katsoa varsinaisen videon. Yleisenä tavoitteena DiMaS-järjestelmässä on ratkaista, miten turvattoman P2P-verkon yli voidaan siirtää materiaalia siten, että teoksen tekijä voi jatkossakin saada korvauksen tekemästään työstä.

Haaste 3. Miten suositukset tuotetaan monilähteistä tietoa hyödyntäen?

Tässä tutkielmassa esiteltyjen järjestelmien perusteella monilähteisessä suosittelussa hyödynnetään samoja tekniikoita kuin luvussa 2 esitellyissä perinteisissä suosittelumenetelmissä.

Monilähteisissä suosittelujärjestelmissä suositusten tuotanto jakaantuu kahteen vaiheeseen, joista ensimmäisen aikana tapahtuu tietojen keruu, mahdollinen esiprosessointi ja muunto yhteismitalliseen muotoon. Toisessa vaiheessa tehdään varsinainen

suositusten tuottaminen ensimmäisessä vaiheessa kerättyyn tietoon perustuen. Mahdollinen esiprosessointi voi sisältää esimerkiksi Zieglerin ja Golbeckin tapaan käyttäjäprofiilien määrän vähentämistä ja profiilien tarkkuustason supistamista, jotta ne limittyisivät paremmin toisiinsa. Esiprosessointi parantaa suosittelun onnistumistodennäköisyyttä.

Tähän mennessä semanttisen webin tietoja hyödyntäviä monilähteisiä suosittelujärjestelmiä on toteutettu yhteistoiminnallisena suositteluna sekä tietämuspohjaisena suositteluna. Jatkotutkimukselle jää sijaa selvittää muiden luvussa 2 esiteltyjen suosittelumenetelmien käyttöä semanttisen webin tietoihin perustuen.

Jatkotutkimukselle jää sijaa myös selvittää, missä määrin suositusten tuotannon hajauttaminen semanttisen webin tekniikoita hyödyntäen olisi mielekästä. Käytännössä tämä tarkoittaisi sitä, että eri suosittelujärjestelmät kommunikoisivat toistensa kanssa semanttisen webin tekniikoiden avulla. Esimerkiksi Ontodella tukee periaatteessa tällaista hajauttamista, koska suositukset tuotetaan RDF-muodossa, jolloin jokin muu järjestelmä voisi jatkojalostaa suosituksia eteenpäin.

Haaste 4. Miten monilähteiset suositukset esitetään käyttäjälle?

Edellä kuvattiin MuseoSuomi-järjestelmän tapa esittää semanttisen webin tietämuspohjaisia suosituksia käyttäjälle. Kuhunkin suositukseen liittyy yksinkertainen selitys siitä, miten suosituksen kohde liittyy nykyiseen kohteeseen. Selitysteksti on mahdollista tuottaa, koska suositusten tuotannossa on käytössä ontologiaan tallennettua tietämystä käsitteiden välisistä suhteista.

Lisäksi edellä esiteltiin MuseoSuomen moninäköymähakua, joka tarjoaa käyttäjälle tavan hakea semanttisesti kuvailtua tietoa useasta eri näkökulmasta. Yksi mielenkiintoinen jatkokehitysidea olisi se, että MetaLens-tyyppiseen, useita erilaisia ehtoja (esimerkiksi kategoriaehdot ja arvoalue-ehdot sekä painotukset) sisältävään haku-

käyttöliittymään sovellettaisiin moninäkömahakua. Etuna olisi se, että hakutulosta rajattaisiin jatkuvasti käyttäjän tekemien valintojen perusteella, eikä kuten MetaLensissä, jossa käyttäjä täyttää ensin ison joukon valintoja hakulomakkeesta, ja vasta tämän jälkeen hän näkee, miten ehdot vaikuttivat tulokseen. Hierarkkiset näkymät voisivat myös jäsentää sisältöä paremmin kuin MetaLensin yksitasoiset valintalistat. (Myös Schafer on kehittänyt MetaLensin myöhempään versioon dynaamisen hakutoiminnon [Sch05].)

7 Yhteenveto

Tutkielman tavoitteena oli pohtia, mitä uusia mahdollisuuksia suosittelujärjestelmille olisi semanttisen webin tekniikoista sekä miten semanttinen web hyötyisi suosittelujärjestelmistä. Tutkielman puitteissa vaikuttaa siltä, että suosittelujärjestelmät ja semanttisen webin tekniikat täydentävät hyvin toisiaan. Suosittelujärjestelmien kannalta semanttinen web tarjoaa keinon yhdistää useista lähteistä tulevaa tietoa, joita voidaan hyödyntää suositusten tuottamiseen. Koska semanttinen web muodostaa valtavan maailmanlaajuisen tietokokoelman, on semanttisessa webissä puolestaan tarvetta suosittelujärjestelmille, jotta käyttäjä saisi parhaimman hyödyn tarjolla olevista tiedoista.

Käytännön esimerkkinä tutkielmassa esiteltiin MuseoSuomi-järjestelmään kehitetty suosittelujärjestelmä Ontodella, joka linkittää loogisten sääntöjen avulla eri lähteistä kerätyn aineiston yhtenäiseksi WWW-sivustoksi. Sääntöpohjainen suosittelu osoitautui toimivaksi tavaksi linkittää semanttisesti kuvattuja sisältöjä toisiinsa, mikäli aineisto, ontologiat ja säännöt vain pystytään tuottamaan.

Automaattisesti toimiva sääntöpohjainen linkittäminen vähentää käsin tehtävää työtä yksittäisen kohteen (esimerkiksi museoesine) osalta, jolloin seurauksena voi olla sekä kustannussäästöjä että mahdollisuus luoda palveluita, joihin ei muuten löytyisi resursseja. Erityisesti isot WWW-sivustot ja -portaalit hyötyvät tällaisista tekniikoista, joten on jossain määrin todennäköistä, että automaattiset linkitysmenetelmät yleistyvät tulevaisuudessa.

Sääntöpohjaisesti linkittämisestä syntyy toisaalta myös lisätyötä, koska kohteet on kuvattava (annotoitava) yhteisten ontologioiden avulla. MuseoSuomen osalta annotointi pystyttiin kuitenkin automatisoimaan kohtuullisen pitkälle, koska käytettävä sisältö oli kuvattu kohtuullisen yhdenmukaista terminologiaa käyttäen. Annotointeja voidaan hyödyntää sääntöpohjaisten linkkien lisäksi myös esimerkiksi semanttisten

hakuominaisuuksien tarjoamisessa.

Vaikka Ontodella toteuttaa onnistuneesti MuseoSuomen tarpeet, on myös jatkokehityskohteita helppo osoittaa. Sääntöjen tuottaminen olisi syytä tehdä helpommaksi, jotta myös ohjelmointitaidottomat sisältöasiantuntijat pystyisivät tuottamaan ja muokkaamaan linkitysperiaatteita. Lisäksi olisi ratkaistava, miten suositukset saataisiin mielekkääseen paremmuusjärjestykseen, koska nykyisellään suositukset ovat satunnaisessa järjestyksessä. Monilähteen suosittelun kannalta mielenkiintoinen haaste olisi myös se, miten uusia tietolähteitä voisi ottaa käyttöön jopa suorituksen aikana siten, että esimerkiksi käyttäjä voisi antaa haluamansa tietolähteen URL-osoitteen Ontodellalle, jonka jälkeen tämä tieto vaikuttaisi tuotettaviin suosituksiin.

Lähteet

- Ama Amazon.com. <http://www.amazon.com/>. [30.1.2006].
- AvH04 Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer*. Massachusetts Institute of Technology, 2004.
- BHK98 John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI*, pages 43–52. Morgan Kaufmann Publisher, July 1998.
- BL00 Tim Berners-Lee. Semantic web on xml -presentation. XML 2000 Washington DC conference, December 2000. <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>.
- BLHL01 Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
- BS97 Marko Balabanovic and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, March 1997.
- Bur00 Robin Burke. Knowledge-based recommender systems. In A. Kent, editor, *Encyclopedia of Library and Information Systems*, volume 69, Supplement 32. Marcel Dekker, New York, 2000.
- Bur02 Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modelling and User-Adapted Interaction*, 12(4):331–370, November 2002.
- BYRN99 Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
- DHN03 Peter Dolog, Nicola Henze, and Wolfgang Nejdl. Logic-based open hypermedia for the semantic web. In *Proceedings of 1st International Workshop on Hypertext and Semantic Web, International Hypertext Conference, August 26-31, Nottingham, UK*, 2003.
- Dmo Dmoz.org. <http://www.dmoz.org/>. [30.1.2006].
- Fen01 Dieter Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, 2001.
- GKRT04 R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *International World Wide Web Conference (WWW2004)*, pages 403–412, New York, USA, 2004. ACM Press.

- GNOT92 David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.
- GNP05 Stefania Ghita, Wolfgang Nejdl, and Raluca Paiu. Semantically rich recommendations in social networks for sharing, exchanging and ranking semantic context. In *The Semantic Web - ISWC 2005 - 4th International Semantic Web Conference, ISWC 2005*, pages 293–307, Galway, Ireland, November 6-10 2005. Springer-Verlag. LNCS 3729.
- Goo Google. <http://www.google.com/>. [30.1.2006].
- GRGP01 Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.
- Gru93 Thomas R. Gruber. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5(2):199–220, June 1993.
- GS05 A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *International World Wide Web Conference (WWW2005)*, Chiba, Japan, May 10-14 2005.
- Gut98 Robert H. Guttman. Merchant differentiation through integrative negotiation in agent-mediated electronic commerce. Master's thesis, Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences, 1998. <http://hdl.handle.net/1721.1/9738>.
- Hea00 Marti A. Hearst. Next generation web search: Setting our sites. *Bulletin of the Technical Committee on Data Engineering*, 23(3):38–48, September 2000.
- Her00 Jonathan Lee Herlocker. *Understanding and Improving Automated Collaborative Filtering Systems*. PhD thesis, University of Minnesota, 2000.
- HK01 Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Academic Press, 2001.
- HKR00 Jonathan H. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Computer Supported Cooperative Work*, pages 241–250. ACM, 2000.
- HMS⁺05 Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula. MuseumFinland – Finnish Museums on the Semantic Web. *Journal of Web Semantics, Elsevier*, 2005. Full length version of a paper to be published by the Journal of Web Semantics, <http://www.seco.tkk.fi/publications/>.

- HSV04 Eero Hyvönen, Samppa Saarela, and Kim Viljanen. Application of ontology techniques to view-based semantic search and browsing. In *The Semantic Web: Research and Applications. Proceedings of the First European Semantic Web Symposium (ESWS 2004)*, 2004.
- HTM HyperText Markup Language (HTML) Home Page. <http://www.w3.org/MarkUp/>. [27.1.2006].
- HVVH04 Eero Hyvönen, Arttu Valo, Kim Viljanen, and Markus Holi. A logic-based semantic web html generator – a poor man’s publishing approach. In *Proceedings of WWW2004, Alternate Track Papers and Posters*, New York, USA, May 2004. ACM Press.
- Hyv02 Eero Hyvönen, editor. *Semantic Web Kick-Off in Finland: Vision, Technologies, Research, and Applications*, volume 1 of *HIIT Publications*. Helsinki Institute for Information Technology (HIIT), 2002. <http://www.hiit.fi>.
- II93 B. Joseph Pine II. *Mass Customization*. Harvard Business School Press, Boston, Massachusetts, 1993.
- ISB U.S. ISBN Agency. <http://www.isbn.org/>. [27.1.2006].
- ISR The International Standard Recording Code. <http://www.ifpi.org/isrc/>. [27.1.2006].
- Jav Java.sun.com: The Source for Java Developers. <http://java.sun.com>. [27.1.2006].
- Jen Jena—A Semantic Web Framework for Java. <http://jena.sourceforge.net/>. [30.1.2006].
- KM02 Marja-Riitta Koivunen and Eric Miller. *W3C Semantic Web Activity*, chapter 2, pages 27–43. In Hyvönen [Hyv02], 2002.
- Kru97 Bruce Krulwich. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.
- MHSV04 E. Mäkelä, E. Hyvönen, S. Saarela, and K. Viljanen. Ontoviews—a tool for creating semantic web portals. In *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan*, pages 797–811. Springer-Verlag, Berlin, November 2004.
- Mov MovieLens. <http://movielens.umn.edu>. [30.1.2006].
- Mpe The Reference Website for MPEG! <http://www.mpeg.org/>. [27.1.2006].
- N3 Notation 3. <http://www.w3.org/DesignIssues/Notation3.html>. [27.1.2006].

- Nic97 David M. Nichols. Implicit rating and filtering. In *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering, Budapest, Hungary, 10-12 November 1997, ERCIM*, pages 31–36, 1997.
- Ora Orava - Opetusvideoiden haku- ja suosittelukone. <http://demo.seco.tkk.fi/orava>. [30.1.2006].
- OWL OWL Web Ontology Language. <http://www.w3.org/TR/owl-ref/>. [27.1.2006].
- PBMW98 Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
- PDF Portable Document Format. http://en.wikipedia.org/wiki/Portable_Document_Format. [27.1.2006].
- Por Porkkalan vuokraus 1944-1956. <http://koulut.kirkkonummi.fi/~seppa/porkka/>. [30.1.2006].
- Pro Protégé Ontology Editor. <http://protege.stanford.edu/>. [30.1.2006].
- RDFa Resource Description Framework (RDF). <http://www.w3.org/RDF/>. [27.1.2006].
- RDFb RDF Vocabulary Description Language 1.0: RDF Schema. <http://www.w3.org/TR/rdf-schema/>. [27.1.2006].
- RIS⁺94 P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.
- Rot Rottentomatoes. <http://www.rottentomatoes.com>. [30.1.2006].
- RS04 Tommo Reti and Risto Sarvas. The DiMaS system for authoring communities. In *Proceedings of the 11th Finnish Artificial Intelligence Conference STeP 2004, September 1-3, Vantaa, Finland*, volume 2 of *Conference Series – No 20*, pages 147–161. Finnish Artificial Intelligence Society, 2004.
- RV97 Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, March 1997.
- Sch01 John Benjamin Schafer. *MetaLens: A Framework for Multi-source Recommendations*. PhD thesis, University of Minnesota, 2001.

- Sch05 J. Ben Schafer. Dynamiclens: A dynamic user-interface for a meta-recommendation system. Beyond Personalization, workshop, In conjunction with the 2005 International Conference on Intelligent User Interfaces (IUI 2005), January 9-12 2005.
- SFW83 Gerard Salton, Edward A. Fox, and Harry Wu. Extended boolean information retrieval. *Communications of the ACM*, 26(12):1022–1036, 1983.
- Sid05 Teemu Sidoroff. Semanttiset portaalit. Master's thesis, Helsingin yliopisto, Tietojenkäsittelytieteen laitos, maaliskuu 2005.
- SKR01 J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1/2):115–153, 2001.
- SKR02 J. Ben Schafer, Joseph A. Konstan, and John Riedl. Meta-recommendation systems: User-controlled integration of diverse recommendations. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 43–51, 2002.
- SN00 Ian Soboroff and Charles Nicholas. Collaborative filtering and the generalized vector space model. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 351–353, 2000.
- Sor04 Martti Soramäki. *Informaatioyhteiskunnan teorian, politiikka ja sähköisen viestinnän todellisuus*. Tampere University Press, 2004.
- Squ Squid Web Proxy Cache. <http://www.squid-cache.org/>. [30.1.2006].
- STK⁺92 Raimo Seppänen, Seppo Tiihonen, Martti Kervinen, Reino Korpela, Lassi Mustonen, Anja Haavisto, Marjatta Soininen, and Kiuru Varho. *MAOL-taulukot*. Kustannusosakeyhtiö Otava, 1992.
- SUM SUMO Ontology. <http://ontology.teknowledge.com/>. [27.1.2006].
- Suo Suomi.fi. <http://www.suomi.fi/>. [30.1.2006].
- SW W3C Semantic Web. <http://www.w3.org/2001/sw/>. [27.1.2006].
- SWI SWI-Prolog. <http://www.swi-prolog.org>. [30.1.2006].
- URIA Naming and Addressing: URIs, URLs, ... <http://www.w3.org/Addressing>. [27.1.2006].
- URIB URIs, URLs, and URNs: Clarifications and Recommendations 1.0. <http://www.w3.org/TR/uri-clarification/>. [27.1.2006].

- Usc03 Michael Uschold. Where are the semantics in the semantic web? *AI Magazine*, 24(3):23–36, 2003.
- WAP WAP Forum Releases.
<http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html>. [27.1.2006].
- Whe00 Malcolm Wheatley. Jeff bezos takes everything personally. *CIO Magazine*, August 2000. <http://www.cio.com/archive/080100/bezos.html>.
- WSW03 Jan Wielemaker, Guus Schreiber, and Bob Wielinga. Prolog-based infrastructure for RDF: performance and scalability. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *The Semantic Web - Proceedings ISWC'03, Sanibel Island, Florida*, pages 644–658, Berlin, Germany, october 2003. Springer Verlag. LNCS 2870.
- XML Extensible Markup Language (XML). <http://www.w3.org/XML/>. [30.1.2006].
- Yah Yahoo. <http://www.yahoo.com/>. [30.1.2006].
- YSO Yleinen Suomalainen Ontologia (YSO). <http://www.seco.tkk.fi/ontologies/yso/>. [27.1.2006].
- ZG05 Cai-Nicolas Ziegler and Jennifer Golbeck. Investigating correlations of trust and interest similarity - do birds of a feather really flock together? *Journal of Artificial Intelligence Research*, 2005. submitted.
- Zie04 Cai-Nicolas Ziegler. Semantic web recommender systems. In *Current Trends in Database Technology - EDBT 2004 Workshops, EDBT 2004 Workshops PhD, DataX, PIM, P2P&DB, and ClustWeb, Heraklion, Crete, Greece, March 14-18, 2004, Revised Selected Papers*, volume 3268 of *Lecture Notes in Computer Science*, pages 78–89. Springer, 2004.

Liite 1. Liittyy tapahtumaan -suositteelusääntö

```

related_by_event( Subject, Target, Explanation ) :-
% define a few property URIs
ItemTypeProperty =
  'http://www.cs.helsinki.fi/seco/ns/2004/03/artifacts#item_type',
ItemTypeToEventRelatingProperty =
  'http://www.cs.helsinki.fi/seco/ns/2004/03/mapping#related_to_event',

% check that both URIs correspond in fact to artifacts
isArtifact(Subject),

% find all the item types the subject item belongs to
rdf(Subject, ItemTypeProperty, SubjectItemType),
rdfs_transitive_subClassOf(SubjectItemType,SubClassOfSubjectItemType),

% find all the events any of those item types are related to
rdf(SubClassOfSubjectItemType, ItemTypeToEventRelatingProperty,Event),
% and events they include or are part of
(
  rdfs_transitive_subClassOf(Event, SubOrSuperClassOfEvent),
  DescResource=SubOrSuperClassOfEvent;
  % or
  rdfs_transitive_subClassOf(SubOrSuperClassOfEvent, Event),
  DescResource=Event;
),

% find all item types related to those events
rdf(TargetItemType, ItemTypeToEventRelatingProperty, SubOrSuperClassOfEvent),
% and all their superclasses
rdfs_transitive_subClassOf(SuperClassOfTargetItemType, TargetItemType),

% don't make uninteresting links between items of the same type
SuperClassOfTargetItemType \= SubjectItemType,
not(rdfs_transitive_subClassOf(SuperClassOfTargetItemType, SubjectItemType)),
not(rdfs_transitive_subClassOf(SubjectItemType, SuperClassOfTargetItemType)),

% finally, find all items related to the linked item types
rdf(Target, ItemTypeProperty, SuperClassOfTargetItemType),

% ensure, that the subject is not the target also
Subject \= Target,

% check that the Target is an artifact
isArtifact(Target),

% create the description for the recommendation
list_labels([DescResource], RelLabel),
Explanation=[commonResources(DescResource), label(fi:RelLabel)].

```

Liite 2. Ontodellan RDF/XML-vastaus

Luettavuuden parantamiseksi alla olevaan XML-dokumenttiin on lisätty rivinvaihtoja ja sisennyksiä joita alkuperäisessä dokumentissa ei ollut.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF xmlns:bm="http://www.w3.org/2002/01/bookmark#"
  xmlns:ogt="http://www.cs.helsinki.fi/group/seco/ns/2004/03/ontogator#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:swhg="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-sewehgrius#"
  xmlns:fms="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#"
  xmlns:mao="http://www.cs.helsinki.fi/group/seco/ns/2004/03/30-mao#"
  xmlns:masa="http://www.cs.helsinki.fi/group/seco/ns/2004/03/30-masa#"
  xmlns:maoryhma="http://www.cs.helsinki.fi/group/seco/ns/2004/03/30-maoryhma#"
  xmlns:terms="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-terms#"
  xmlns:ajat="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-ajat#"
  xmlns:kokoelmat="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-kokoelmat#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:od="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-ontodella#">

<!-- metadata about ontodella -->
  <ogt:CategoryBase rdf:about="">
    <owl:versionInfo>silent_squirrel</owl:versionInfo>
    <od:ontodellaCvsId>$Id: ontodella-vastaus.tex,v 1.1 2005/11/17 11:35:40 kimvilja
Exp $</od:ontodellaCvsId>
    <od:sewehgrius_dataid>silent_squirrel</od:sewehgrius_dataid>
    <od:sewehgrius_ruleid>$Id: fmsweb_sewehgrius_rules.pl,v 1.42 2004/09/07 12:05:05
jiemakel Exp $</od:sewehgrius_ruleid>
    <od:replyCreationTime>Tue Nov 15 20:35:10 2005</od:replyCreationTime>

    <od:ontodellaHost>seco-museo.hut.fi</od:ontodellaHost>
  </ogt:CategoryBase>

  <rdf:Description
    rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#ECM_1861_18">
    <swhg:relations rdf:parseType="Collection">
      <swhg:RelationGroup>
        <swhg:relationDescription
          rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-deduced/
relation#same_time_of_manufacture"/>
        <rdfs:label xml:lang="fi">Sama valmistusaika</rdfs:label>
        <rdfs:label xml:lang="en">Same time of manufacture</rdfs:label>
        <swhg:relations rdf:parseType="Collection">
          <swhg:RelationGroup>
            <rdfs:label>1930-1960</rdfs:label>
            <swhg:relationDescription rdf:parseType="Collection">
              <swhg:commonResource>
                <rdfs:label>1930</rdfs:label>
              </swhg:commonResource>
              <swhg:commonResource>
                <rdfs:label>1960</rdfs:label>
              </swhg:commonResource>
            </swhg:relationDescription>
          </swhg:relationDescription>
          <swhg:relations rdf:parseType="Collection">
            <swhg:relation
              swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_1861_1"
              swhg:weight="1">
              <rdfs:label xml:lang="fi">merkinantolippu A:merkinantolippu</rdfs:label>
            </swhg:relation>
            <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_2696_1"
              swhg:weight="1">
              <rdfs:label xml:lang="fi">liitupakkaus:taululiitupakkaus</rdfs:label>
            </swhg:relation>
            <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_3531_1"
              swhg:weight="1">
```

```

        <rdfs:label xml:lang="fi">kori:pajukori</rdfs:label>
    </swhg:relation>
    <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_3572_102"
        swhg:weight="1">
        <rdfs:label xml:lang="fi">joulukoriste:kuusenkoriste</rdfs:label>
    </swhg:relation>
    <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_3572_127"
        swhg:weight="1">
        <rdfs:label xml:lang="fi">hiuspanta</rdfs:label>
    </swhg:relation>
    <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_3572_22"
        swhg:weight="1">
        <rdfs:label xml:lang="fi">käsityökehys:pöytäliinan valmistukseen käytettävä
kehys</rdfs:label>
    </swhg:relation>
    <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_3572_95"
        swhg:weight="1">
        <rdfs:label xml:lang="fi">lompakko:rahapussi</rdfs:label>
    </swhg:relation>
</swhg:relations>
</swhg:RelationGroup>
</swhg:relations>
</swhg:RelationGroup>

<swhg:RelationGroup>
  <swhg:relationDescription
    rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-deduced/
relation#same_place_of_manufacture"/>
    <rdfs:label xml:lang="fi">Sama valmistuspaikka</rdfs:label>
    <rdfs:label xml:lang="en">Same place of manufacture</rdfs:label>
    <swhg:relations rdf:parseType="Collection">
      <swhg:RelationGroup>
        <rdfs:label xml:lang="fi">Leppävaara</rdfs:label>
        <swhg:relationDescription rdf:parseType="Collection">
          <swhg:commonResource rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-paikat#leppavaara">
            <rdfs:label xml:lang="fi">Leppävaara</rdfs:label>
          </swhg:commonResource>
        </swhg:relationDescription>
        <swhg:relations rdf:parseType="Collection">
          <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_1861_1"
            swhg:weight="1">
            <rdfs:label xml:lang="fi">merkinantolippu A:merkinantolippu</rdfs:label>
          </swhg:relation>
          <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_3015_33"
            swhg:weight="1">
            <rdfs:label xml:lang="fi">esiliina:esiliina</rdfs:label>
          </swhg:relation>
          <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_949_1" swhg:weight="1">
            <rdfs:label xml:lang="fi">maljakko:maljakko</rdfs:label>
          </swhg:relation>
        </swhg:relations>
      </swhg:RelationGroup>
    </swhg:relations>
  </swhg:RelationGroup>

<swhg:RelationGroup>
  <swhg:relationDescription
    rdf:resource="http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-deduced/
relations#related_to_an_event"/>
    <rdfs:label xml:lang="fi">Liittyy tapahtumaan</rdfs:label>
    <rdfs:label xml:lang="en">Related to an event</rdfs:label>
    <swhg:relations rdf:parseType="Collection">
      <swhg:RelationGroup>
        <rdfs:label xml:lang="fi">merenkulku</rdfs:label>
      </swhg:RelationGroup>
    </swhg:relations>
  </swhg:RelationGroup>

```

```

    <swhg:relationDescription rdf:parseType="Collection">
      <swhg:commonResource rdf:about="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
30-masa#merenkulku">
        <rdfs:label xml:lang="fi">merenkulku</rdfs:label>
      </swhg:commonResource>
    </swhg:relationDescription>
    <swhg:relations rdf:parseType="Collection">
      <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_3594_155">
        swhg:weight="1">
          <rdfs:label xml:lang="fi">pelastusliivi:korckkitaytteinen
pelastusliivi</rdfs:label>
        </swhg:relation>
      <swhg:relation swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#ECM_3594_156">
        swhg:weight="1">
          <rdfs:label xml:lang="fi">pelastusliivi:korckkitaytteinen pelastusliivi</rdfs:label>
        </swhg:relation>
      <swhg:relation
        swhg:target="http://www.cs.helsinki.fi/group/seco/ns/2004/03/
18-esinekortti#LahtiLKM_LHM_LHM_ES_97070_279">
        swhg:weight="1">
          <rdfs:label xml:lang="fi">Pusero, lapsen:neulepusero, merimiespusero,
tytön</rdfs:label>
        </swhg:relation>
      </swhg:relations>
    </swhg:RelationGroup>
  </swhg:relations>
</swhg:RelationGroup>
</rdf:Description>
</rdf:RDF>

```