TEKNILLINEN KORKEAKOULU
Sähkö- ja tietoliikennetekniikan osasto

Onni Valkeapää

Verkkoresurssien ontologiaperustainen annotointi

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-insinöörin tutkintoa varten Espoossa 24.8.2006

Työn valvoja Professori Eero Hyvönen

Teknillinen korkeakoulu Diplomityön tiivistelmä

Tekijä: Onni Valkeapää

Työn nimi: Verkkoresurssien ontologiaperustainen annotointi

Päivämäärä: 24.08.2006 Sivumäärä: 70

Osasto: Sähkö- ja tietoliikennetekniikan osasto
Professuuri: AS-75 Viestintätekniikka

Työn valvoja: Professori Eero Hyvönen

Tiivistelmäteksti:

Tässä diplomityössä tutkittiin semanttisen webin ontologiaperustaista annotointia ja kehitettiin menetelmä sekä järjestelmä webissä olevien dokumenttien annotointiin. Ontologiaperustaisella annotoinnilla tarkoitetaan webissä olevien resurssien, kuten www-sivujen kuvailua ontologisia käsitteitä käyttäen. Ontologialla viitataan formaalisti määriteltyyn abstraktiin tietomalliin, joka kuvailee jotain maailman ilmiötä ja siihen liittyviä olennaisia käsitteitä. Annotaatioiden avulla erilaiset tietojärjestelmät pystyvät käyttämään älykkäitä tekniikoita tiedon tulkitsemiseksi ja hyödyntämiseksi. Annotaatiot voivat toimia pohjana esimerkiksi erilaisten semanttisten portaalien ja hakupalvelujen muodostamisessa.

Työssä selvitettiin aluksi, minkälaisia annotaatioita webin yhteydessä käytetään ja minkälaisilla menetelmillä niitä tuotetaan. Lisäksi esiteltiin annotaatioiden tuottamiseen kehitettyjä sovelluksia ja tutkittiin minkälaisia ominaisuuksia ne tarjoavat. Annotaatio-ohjelmistojen ei havaittu soveltuvan kovin hyvin sellaisille käyttäjille, joilla ei ole asiantuntemusta semanttiseen webin tekniikoista. Lisäksi sovelluksissa havaittiin puutteita annotointiprosessin ja annotaatioiden jakamisessa eri sisällöntuottajien välillä, mitä pidetään tärkeänä semanttisen webin sisällönkuvailussa.

Semanttisen webin annotointia tukevalle sovellukselle määritettiin yleiset vaatimukset, joita ovat monimutkaisten ontologioiden piilottaminen käyttäjältä, annotoinnin hajauttaminen, skeemojen käyttö sekä selainpohjaisuus. Näiden pohjalta laadittiin annotointisovellus *Saha*, jonka avulla voidaan muodostaa ontologioihin perustuvia, webissä olevia dokumentteja kuvaavia annotaatioita. Sahalla suoritettavaa sisällönkuvailua on testattu FinnONTO-projektissa. Ohjelman testeistä saadut tulokset osoittavat tuetun annotointimallin toimivan odotetulla tavalla, mutta jatkokehityksen olevan yhä tarpeellista.

Avainsanat: semanttinen web, annotointi, metadata, skeema, ontologia

Author: Onni Valkeapää

Name of the Thesis: Ontology–based Annotation of Web Resources

Date: 24.08.2006 Number of pages: 70

Department: Department of Electrical and Communications Engineering
Professorship: AS-75 Media Technology

Supervisor: Professor Eero Hyvönen

Abstract:

The focus of this master's thesis was to study the ontology-based annotation for the Semantic Web and to develop a method and a system for the annotation of web-resources. Ontology-based annotation refers to describing web-resources, such as web-pages, using ontological concepts. Ontology is a formal, abstract model of some phenomenon in the world which identifies the relevant concepts of that phenomenon. Using annotations based on ontologies, different information systems are able to use intelligent methods to process and interpret data. Annotations can be used e.g. to create semantic portals and search services.

The work begun by studying different kinds of annotations that exist on the web and methods used to produce them. After that, applications designed for annotation were surveyed. The applications were not found very well-suited for the users that are not experts in the field of the Semantic Web. In addition to this, the applications were lacking a proper support for collaborative annotation and sharing of annotations created by different annotators. These features are essential in describing the contents of the Semantic Web.

General requirements for an application supporting the annotation were identified. The requirements were 1) hiding complex ontologies from the user, 2) distribution of annotation, 3) use of schemas and 4) implementation as a browser-based application. According to these requirements, an annotation tool *Saha*, that can be used to create ontology-based descriptions of web-documents, was implemented. The Saha is on trial use in the metadata creation for the FinnONTO-project. Initial feedback from the tests have shown that the annotation method supported in Saha works as expected but further experimenting is still needed.

Keywords: semantic web, annotation, metadata, schema, ontologies

Alkusanat

Tämä diplomityö tehtiin Teknillisessä korkeakoulussa viestintätekniikan laboratorion semanttisen laskennan tutkimusryhmässä. Haluan kiittää professori Eero Hyvöstä työni kannustavasta ohjaamisesta ja hyvistä neuvoista. Lisäksi haluan kiittää kaikkia läheisiäni saamastani tuesta sekä työtoveriani Olli Almia työhön liittyvistä hyödyllisistä keskusteluista ja kommenteista.

Espoon Matinlahdessa 24.8.2006

Onni Valkeapää

Sisällysluettelo

LIITTEET

Lyhenteet

HTML	*HyperText Markup Language*. Merkintäkieli, joka on suunniteltu www-sivujen esittämiseen.
HTTP	*HyperText Transfer Protocol*. Protokolla, jota käytetään tiedon välittämiseen www:ssä.
OWL	*Web Ontology Language*. W3C:n suositus webin ontologiakieleksi.
RDF	*Resource Description Framework*. W3C:n suositus webissä olevan metatiedon kuvaamistavasta.
RDFS	*Resource Description Framework Schema*. RDF:llä määritelty sanastonkuvauskieli.
SQL	*Structured Query Language*. Yleinen kieli, jota käytetään relaatiotietokantojen luomisessa, muokkaamisessa ja tiedonhaussa
URI	*Universal Resource Identifier*. Merkkijono, jonka avulla voidaan kertoa esimerkiksi tietyn resurssin paikka (URL) tai yksikäsitteinen nimi (URN).
URL	*Universal Resource Locator*. Internetissä sijaitsevan resurssin paikka eli osoite.
URN	*Universal Resource Name*. URI-skeema, jonka avulla voidaan yksikäsitteisesti nimetä jokin resurssi. URN:lla nimetty resurssi voidaan sitoa johonkin fyysiseen paikkaan URL:n avulla.
W3C	*World Wide Web Consortium*. Internetin teknologioita standardoiva kansainvälinen järjestö.
XHTML	*Extensible HyperText Markup Language*. HTML:a vastaava merkintäkieli, jonka syntaksi on täsmällisemmin määritelty.
XML	*Extensible Markup Language*. W3C:n suositus yleiskäyttöiseksi merkintäkieleksi, jonka avulla voidaan mm. määritellä muita merkintäkieliä.

1 JOHDANTO

1.1 Tutkimuksen tausta

Semanttisella webillä viitataan World Wide Webin rinnalle rakennettavaan tietokerrokseen, jossa tieto kuvaillaan siten, että koneet pystyvät käsittelemään sitä nykyistä tehokkaammin ja käyttämään älykkäitä tekniikoita tiedon hyödyntämiseksi (Antoniou & van Harmelen, 2004). Jotta tällainen tietokerros voitaisiin rakentaa nykyisen webin yhteyteen, tulee webissä olevia ja sinne tuotettavia tietosisältöjä kuvailla semanttisesti, eli liittää tietoon sen merkityksestä kertovaa metatietoa. Annotoinniksi kutsuttua prosessia, jossa metatietoa eli annotaatioita tuotetaan, voidaan pitää yhtenä semanttisen webin tämän hetken tärkeimmistä haasteista.

Semanttisen webin menestykseen vaikuttaa ratkaisevalla tavalla se, miten hyvin webin resursseja kuvaavia annotaatioita pystytään tuottamaan. Webin nykyisessä kehitysvaiheessa suurin osa tiedosta on kuvattu luonnollista kieltä käyttäen ja sitä voidaan pitää yhtenä semanttisen webin palveluiden kehittämisen merkittävimmistä esteistä (Dill ym. 2003). Annotaatioita koskevat ongelmat liittyvät sekä annotaatioiden tuottamiseen että niiden varsinaiseen hyödyntämiseen. Annotaatioita hyväksi käyttäviä tehokkaita sovelluksia ei ole kannattanut annotaatioiden puuttuessa kehittää, eikä toisaalta annotaatioitakaan ole tuotettu riittävästi, koska niitä hyödyntäviä sovelluksia ei ole ollut. Tätä semanttisen webin kehityksen kannalta ongelmallista kierrettä on entisestään vaikeuttanut Internetin valtava tietomäärä, jonka semanttiseen kuvailuun on yritetty löytää tehokkaita keinoja.

Semanttisen webin annotointia varten tarvitaan sovelluksia, joilla voidaan tuottaa koneiden tulkittavaksi tarkoitettua webin resursseja kuvaavaa metatietoa. Tällaisen tiedon tulee olla standardien mukaan määriteltyä ja sen levittämiseen tarvitaan kansainvälisesti yhteensopiva infrastruktuuri (Fensel ym. 2003). Kun webin tietoa kuvaillaan näiden tavoitteiden mukaisesti, voidaan helpommin lähestyä semanttiselle webille asetettua tavoitetilaa, jossa erilaiset sisällöt mahdollisimman saumattomasti kytkeytyvät toisiinsa merkitystensä kautta ja tarjoavat webin käyttäjille uuden sekä ennen kaikkea monipuolisemman näkymän webissä olevaan tietoon. Koska annotaatiot ovat avainasemassa semanttisen webin toteutuksessa, tulee niiden tuottamista tukea mahdollisimman monipuolisesti sekä webin sisällöntuotannossa että jo olemassa olevien sisältöjen rikastamisessa. Jotta mahdollisimman monella webin käyttäjällä olisi mahdollisuus tuottaa tietoa tiedosta, tulee tarjolla olla helppokäyttöisiä annotointityövälineitä, jotka tukevat semanttisten annotaatioiden luomista.

1.2 Tutkimuksen tavoitteet ja rajaus

Tämän diplomityön tavoitteena on tutkia verkossa olevien dokumenttien ontologiapohjaista annotointia sekä selvittää minkälaisilla sovelluksilla annotaatioita

voidaan tuottaa. Tutkimuksen pohjalta tavoitteena on kehittää menetelmä annotaatioiden hajautettuun tuottamiseen, ja toteuttaa uusi helppokäyttöinen web-sovellus, joka osaltaan parantaa nykyisissä sovelluksissa havaittavia puutteita.

Tutkimuksen teoriaosassa käsitellään lyhyesti semanttisen webin annotoinnin yhteydessä käytettäviä automaattisia tiedoneristämismenetelmiä, mutta niiden soveltaminen kehitetyssä annotointimenetelmässä rajataan työn ulkopuolelle. Työssä ei myöskään käsitellä tarkemmin annotaatioiden hyödyntämistä eikä annotaatioiden pohjalta muodostettavia palveluita.

1.3 Tutkimuksen rakenne

Tutkimus koostuu seitsemästä luvusta. Luvussa 2 käsitellään työn kannalta keskeisiä semanttisen webin käsitteitä ja perusteknologioita, jotka toimivat pohjana seuraavissa luvuissa käsiteltävälle annotoinnille. Luvussa 3 perehdytään yleisesti semanttisen webin annotaatioihin sekä niiden tuottamisessa käytettäviin menetelmiin. Luvussa 4 esitellään olemassa olevia annotaatiotyövälineitä, vertaillaan niiden ominaisuuksia ja käydään läpi työvälineille asetettavia vaatimuksia. Luvussa 5 kuvaillaan Saha-annotaatiosovellus ja luvussa 6 arvioidaan saatuja tuloksia sekä esitellään jatkokehitysehdotuksia. Luku 7 on yhteenveto työstä.

2 SEMANTTINEN WEB

Tässä luvussa kerrotaan lyhyesti semanttisen webin perusteknologioista, jotka perustuvat RDF-pohjaisiin (Resource Description Framework) merkkauskieliin. *Semanttisella webillä* tarkoitetaan nykyisen webin[1] perusinfrastruktuurin päälle lisättyä metatietokerrosta, jossa kuvaillaan webissä olevien resurssien välisiä riippuvuuksia ja ominaisuuksia. Kuvailu tapahtuu yhteisesti sovittujen standardien mukaan ja kuvaukset on etupäässä tarkoitettu tietokoneiden käsiteltäviksi. Luvussa käsitellään lisäksi webissä olevan tiedon semanttisessa kuvailussa käytettäviä ontologioita sekä sitä, miten semanttisessa webissä viitataan erilaisiin resursseihin, kuten dokumentteihin.

2.1 Semanttisen webin perusteknologiat

2.1.1 Yleistä

Tällä hetkellä webissä olevien dokumenttien hallitseva merkitsemiskieli on HTML (HyperText Markup Language), jota käytetään pääasiassa dokumenttien ulkoasun kuvailuun siten, että ihmisten on helppo lukea niitä. Ratkaisu on tietokoneiden näkökulmasta huono, koska ulkoasun kuvailuun tarkoitettu dokumentti sisältää vähän tai ei ollenkaan varsinaista sisältöä kuvailevaa *metatietoa*[2]. Webin hakukoneet, kuten Google, yrittävät parhaansa mukaan tulkita HTML-dokumenttien sisältöä ja poimia niistä hakujen kannalta oleellisia asioita. Vaikka nykyisen webin päälle voitaisiin rakentaa entistä älykkäämpiä agentteja[3], jotka osaisivat paremmin tulkita webistä löytyvää ihmisille kuvailtua tietoa, on semanttisen webin kehityksessä pyritty lähestymään ongelmaa agenttien sijaan etupäässä dokumenttien näkökulmasta. Kun dokumentit sisältävät ulkoasun formatointiin liittyvän tiedon lisäksi myös sisältöä kuvailevaa metadataa koneiden käsiteltäväksi tarkoitetussa muodossa, on niiden prosessointi tietokoneilla huomattavasti helpompaa. (Antoniou & van Harmelen, 2004)

2.1.2 Resource Description Framework -kieli (RDF)

RDF on kieli (Manola & Miller, 2004), jonka avulla voidaan esittää tietoa webissä olevista resursseista. Resurssilla tarkoitetaan mitä tahansa asiaa, johon voidaan viitata URI-tunnisteella (RFC 3986, 2005). Tyypillisiä esimerkkejä resursseista ovat elektroninen

---

[1] Sanalla *web* viitataan World Wide Webiin, josta käytetään myös lyhennettä www. Web on Internetin päälle rakennettu hypertekstijärjestelmä, jossa käyttäjä voi tietokoneellaan olevalla web-selaimella hakea tietoja web-palvelimilta. Webissä oleva tieto koostuu joukosta erilaisia dokumentteja, joihin viitataan niiden URI-tunnisteilla (Universal Resource Identifier).

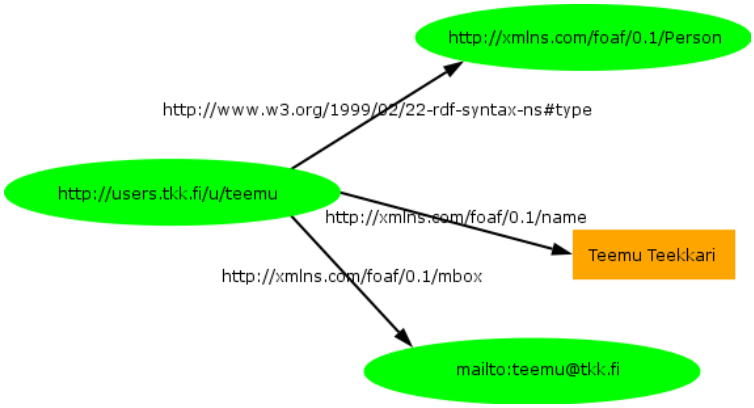
[2] *Metatieto* on tietoa kuvailevaa tietoa. Metatieto voi olla tietoa esimerkiksi jonkun dokumentin teknisistä ominaisuuksista, mutta semanttisen webin yhteydessä sillä viitataan yleensä dokumentin sisältöä ja sen merkitystä kuvailevaan tietoon.

[3] *Agentilla* viitataan tässä yhteydessä tietokoneohjelmaan, joka suorittaa käyttäjän puolesta jonkin toiminnon. Agentti voi esimerkiksi automaattisesti etsiä Internetissä tietoja jostain halutusta aihepiiristä.

dokumentti, kuva, palvelu tai kokoelma muita resursseja. Myös abstraktit käsitteet, kuten matemaattisen yhtälön muuttujat tai asioiden välisiä suhteita kuvaavat käsitteet voivat olla resursseja. RDF-kieli on suunniteltu erityisesti webin resursseihin liittyvän metadatan, kuten esimerkiksi www-sivun otsikon, tekijän ja muokkauspäivämäärän esittämiseen. RDF:n avulla voidaan kuitenkin yhtä lailla kuvata mitä tahansa asiaa, joka voidaan identifioida webissä, vaikka sitä ei voitaisikaan suoraan hakea sieltä. RDF on ensisijaisesti tarkoitettu tilanteisiin, jossa informaatioita ei näytetä ihmisille, vaan käsitellään tietokoneilla. Se tarjoaa yhtenäisen kehyksen esittää tällaista tietoa siten, että sen merkitys on sisällöllisesti yhtyeentoimiva, kun tietoa välitetään sovellukselta toiselle.

RDF perustuu resurssien tunnistamiseen URI:lla ja niiden kuvailuun erilaisilla ominaisuuksilla sekä ominaisuuksien arvoilla. Näiden avulla resursseista voidaan esittää yksinkertaisia *lauseita* (statement). Lauseet voidaan kuvata graafisesti solmuilla ja niitä yhdistävillä kaarilla, jotka esittävät resursseja, ominaisuuksia sekä niiden arvoja.

Kuvassa 1 on Manolaa & Milleriä (2004) mukaillen kuvattu RDF-verkko, jossa joukolla lauseita kuvataan seuraavaa: ”on olemassa henkilö, joka tunnistetaan URI:lla http://users.tkk.fi/u/teemu, jonka nimi on Teemu Teekkari ja jonka sähköpostiosoite on teemu@tkk.fi”.



Kuva 1. *RDF-verkko*

Kuvasta 1 nähdään, että RDF käyttää URI:a identifioimaan:

- yksilöitä, kuten *Teemu Teekkari*, joka tunnistetaan URI:lla http://users.tkk.fi/u/teemu
- yksilöiden luokkia, kuten *Henkilö*, joka tunnistetaan URI:lla http://xmlns.com/foaf/0.1/Person
- ominaisuuksia, kuten *sähköpostiosoite*, joka tunnistetaan URI:lla http://xmlns.com/foaf/0.1/mbox
- ominaisuuksien arvoja, kuten *mailto:teemu@tkk.fi* (arvoina voi olla myös merkkijonoja, kuten ”Teemu Teekkari” sekä muita tietotyyppejä, kuten kokonaislukuja, päivämääriä jne.)

RDF-suositus (Beckett & McBride, 2004) määrittelee myös RDF/XML:ksi kutsutun XML-pohjaisen (Extensible Markup Language) syntaksin, jota voidaan käyttää RDF-graafien esittämiseen. Seuraavassa on esitetty kuvassa 1 kuvatut lauseet RDF/XML:lla:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <foaf:Person rdf:about="http://users.tkk.fi/u/teemu">
    <foaf:name>Teemu Teekkari</foaf:name>
    <foaf:mbox rdf:resource="mailto:teemu@tkk.fi"/>
  </foaf:Person>
</rdf:RDF>
```

Resurssien kuvaamisen lisäksi RDF:a voidaan käyttää uusien metamallien määrittelyssä ja sen vuoksi se toimii pohjana muille semanttisen webin metatietokielille. RDF on sovellusalasta riippumaton, koska se jättää käyttäjien tehtäväksi määritellä omat sovelluskohtaiset terminologiat. Tähän voidaan käyttää RDF:lla määriteltyä sanastonkuvauskieltä, RDF Schemaa (RDFS).

2.1.3 RDF Schema sanastonkuvauskieli

RDF:n avulla voidaan ilmaista yksinkertaisia resursseja kuvaavia lauseita käyttäen nimettyjä ominaisuuksia ja arvoja. RDF:n käyttäjien tulee kuitenkin pystyä määrittämään myös lauseissa käytettäviä sanastoja ilmaistakseen sen, että niissä kuvaillaan juuri tietyllä tavalla luokiteltuja resursseja ja ominaisuuksia. Koska pelkkä RDF ei tarjoa keinoja määritellä sovelluskohtaisia luokkia ja ominaisuuksia, käytetään niiden kuvailuun RDF-sanastonkuvauskieltä *RDF Schemaa* (Brickley & Guha, 2004), joka koostuu joukosta laajennuksia RDF:iin.

RDF Schema ei itsessään sisällä sovelluskohtaisia luokkia ja ominaisuuksia, mutta se tarjoaa keinot niiden kuvailemiseen ja tavan osoittaa mitä luokkia ja ominaisuuksia on tarkoitus käyttää toistensa yhteydessä. RDF Schema tarjoaa toisin sanoen tyyppijärjestelmän RDF:lle, jota voidaan joiltain ominaisuuksiltaan verrata olio-ohjelmointikielien, kuten Javan tyyppijärjestelmään. RDF Scheman avulla voidaan esimerkiksi määritellä, että resurssi on jonkin luokan tai useamman luokan ilmentymä. Luokkia ja ominaisuuksia voidaan lisäksi järjestää ylä- ja alaluokkasuhteet määrittäviin hierarkioihin sekä määritellä ominaisuuksille mm. niiden arvoalueeseen liittyviä rajoitteita.

RDF Scheman ominaisuudet on toteutettu joukolla ennalta määritettyjä RDF-resursseja, joilla on omat erityismerkityksensä. Koska RDFS-kuvaukset on määritelty tavallisina RDF-graafeina, niitä voidaan käsitellä myös sellaisilla sovelluksilla, joita ei ole suunniteltu käsittelemään RDF Schemaa. Tällaisissa tapauksissa sovellus käsittelee RDF:n resursseja ja ominaisuuksia normaalilla tavalla, mutta se ei ymmärrä RDF Scheman tarjoamia lisämerkityksiä.

2.1.4 Webin ontologiakieli OWL

OWL (McQuinness & van Harmelen, 2004) on W3C:n (World Wide Web Consortium) suositus webin ontologiakieleksi. Sen avulla voidaan täsmällisesti kuvata sanastoissa (ontologiat) olevien termien merkitys ja niiden välillä vallitsevat suhteet. OWL tarjoaa RDF:a ja RDF Schemaa laajemmat keinot ilmaista merkityksiä ja sen vuoksi se sopii edellisiä monipuolisemmin tietokoneiden käsiteltäväksi tarkoitetun tiedon esittämiseen. OWL on rakennettu RDF:n ja RDF Scheman päälle ja sen kehitystyössä on otettu lähtökohdaksi eurooppalaisten sekä yhdysvaltalaisten tutkimusryhmien yhteistyössä kehittämä DAML+OIL[4]-ontologiakieli.

Ontologiakielien, kuten OWL:n yhtenä perusvaatimuksena on *formaali semantiikka*, joka kuvailee yksikäsitteisesti tiedon merkityksen. Yksikäsitteisyydellä viitataan siihen, että semantiikka ei pohjaudu subjektiivisiin näkemyksiin, eikä se ole avoin erilaisille tulkinnoille. Formaalin semantiikan ansiosta OWL:lla kuvattuun tietoon voidaan kohdistaa päättelyitä: jos esimerkiksi ominaisuus p1 on määritelty ominaisuuden p2 käänteisominaisuudeksi ja tiedetään, että p2 liittää X:n ja Y:n, voidaan päätellä, että Y liittyy X:n ominaisuuden p1 kautta.

Formaali semantiikka on edellytys ontologiakielien toiselle perusvaatimukselle, *päättelytuelle*. Sen avulla voidaan tehdä automaattisesti esimerkiksi edellä kuvattua tapausta muistuttavia päättelyitä. Formaali semantiikka ja päättelytuki toteutetaan yleensä kytkemällä ontologiakieli johonkin loogiseen formalismiin ja käyttämällä automaattisia päättelykoneita, joita on toteutettu kyseiselle formalismille. (Antoniou & van Harmelen, 2004) OWL perustuu erääseen kuvauslogiikkaan[5] (description logic) ja se mahdollistaa erilaisten päättelykoneiden, kuten Racer:in (Haarslev & Möller, 2003) hyödyntämisen.

OWL on jaettu kolmeen ilmaisuvoimaltaan eritasoiseen muotoon, jotka on suunniteltu erityyppisille käyttötapauksille ja käyttäjäryhmille. *OWL Lite* tukee niitä käyttäjiä, jotka tarvitsevat pääasiassa luokitteluhierarkioita ja yksinkertaisia rajoitteita. RDFS:sta tulevien rajoitteiden (domain, range) lisäksi OWL Lite:ssa voidaan määritellä ominaisuuskohtaisia rajoitteita arvoalueille ja sallittujen arvojen määrälle. *OWL DL* on puolestaan suunniteltu käyttäjille, jotka tarvitsevat mahdollisimman suuren ilmaisuvoiman, tinkimättä laskennallisesta eheydestä, yhdistettynä kohtuulliseen laskennalliseen tehokkuuteen. OWL DL:ssa kaikki päättelyt ovat laskettavissa ja laskennan voidaan osoittaa päättyvän äärellisessä ajassa. *OWL Full* on suunnattu käyttäjille, jotka haluavat mahdollisimman suuren ilmaisuvoiman sekä RDF:n syntaktisen vapauden ilman laskennallisia takuita (täydellistä päättelyä ei voida taata).

[4] http://www.w3.org/TR/daml+oil-reference

[5] http://dl.kr.org/

2.2 Ontologiat

Sanan *ontologia* juuret johtavat filosofiaan ja kreikan kieleen, jossa ontologialla viitataan olemassaolon tutkimiseen keskittyvään filosofian osa-alueeseen. Sen pyrkimyksenä on ollut yleisiä termejä käyttäen identifioida olemassa olevia asioita ja tutkia millä tavoin niitä voidaan kuvailla. Viime aikoina sana ontologia on otettu käyttöön myös tietojenkäsittelyn yhteydessä, jossa sille on tosin annettu alkuperäisestä määritelmästä eroava, teknisempi merkitys. (Antoniou & van Harmelen, 2004)

Gruber (1993) määrittelee tietojenkäsittelyn yhteydessä ontologian *formaaliksi, määritelmäksi käsitteistöstä*. Studerin ym. (1998) mukaan käsitteistöllä viitataan tässä jonkin maailmassa olevan ilmiön abstraktiin malliin, johon liittyvät käsitteet on tunnistettu. Formaalilla tarkoitetaan taas sitä, että ontologian tulee olla koneluettava, mikä sulkee pois luonnollisen kielen käytön sen määrittelyssä. Yleisemmin sanottuna ontologia määrittelee termit, joita käytetään tiettyyn aihepiiriin liittyvän tiedon kuvailemisessa (Heflin, 2004). Ontologioiden avulla pyritään kuvailemaan tietokoneille vastaava käsitemalli, jota ihminen käyttää tulkitessaan erilaisista symboleista koostuvia tekstejä.

Tyypillisesti ontologia muodostuu äärellisestä joukosta termejä ja niiden välisistä suhteista. Tällaiset termit ominaisuuksineen määrittelevät tietyn aihealueen keskeiset *käsitteet*, jotka voidaan jaotella erilaisiin luokkiin. Esimerkiksi yliopistomaailmassa keskeisiä käsitteitä ovat professorit, tutkijat, opiskelijat, kurssit jne. Ontologioissa määritellyt *suhteet* koostuvat tyypillisesti luokkien hierarkioista ja muista ominaisuuksista. Hierarkiassa voidaan esimerkiksi määritellä, että luokka *C* on luokan *D* aliluokka, jos jokainen objekti *C*:ssä sisältyy myös luokkaan *D*. Luokkasuhteiden lisäksi ontologiat voivat sisältää tietoa luokkien ominaisuuksista, ominaisuuksien arvorajoituksista ja luokkien sekä ominaisuuksien välisistä loogisista suhteista.

Semanttisen webin yhteydessä ontologiat tarjoavat yhteiskäyttöisen käsitteistön eri aihealueista. Yhteisesti sovittu käsitteistö on välttämätön, jotta pystytään välttämään erilaisista terminologioista johtuvat eroavuudet. Eroavuuksia syntyy, kun samaa termiä käytetään eri sovelluksissa toisistaan poikkeavissa merkityksissä ja toisaalta myös silloin, kun eri termeillä viitataan eri sovelluksissa samaan asiaan. Tällaiset eroavuudet voidaan selvittää kytkemällä eri terminologiat yhteiseen ontologiaan tai määrittämällä suorat viittaukset eri ontologioiden välille. Kummassakin tapauksessa havaitaan, että ontologiat tukevat semanttista yhteentoimivuutta.

Ontologioiden avulla voidaan helpottaa esimerkiksi Internetissä tehtävien hakujen tarkkuutta. Ontologioihin tukeutuvat hakukoneet voivat hakea sivuja, joissa viitataan haluttuun *käsitteeseen*, sen sijaan että haettaisiin sivuja, joissa tietty monimerkityksinen hakusana esiintyy. Tällä tavalla pystytään tekemään ero esimerkiksi jalkapallossa ja talon kunnostuksessa käytettävän sanan ”maali” välillä. Sanojen monimerkityksellisyyden erottamisen lisäksi ontologioita voidaan hyödyntää myös tiedonhaussa käytettyjen käsitteiden yleistämisessä tai tarkentamisessa. Jos hakukoneelle syötetty hakusana ei tuota yhtään vastausta, hakukone voi ontologian perusteella ehdottaa tiedonhakijalle yleisempää

käsitettä. Jos hakutuloksia on liikaa, voi hakukone vastaavasti ehdottaa käyttämään jotain tarkempaa käsitettä.

Tiedonhakijaa voidaan ohjata käyttämään haun kannalta edullisia käsitteitä erilaisilla ontologioihin pohjautuvilla apuvälineillä, kuten hakusanan automaattisella semanttisella täydentämisellä (Hyvönen & Mäkelä, 2006). Siinä hakujärjestelmä ehdottaa vuorovaikutteisesti käyttäjälle sopivia hakutermejä sen perusteella, mitä käyttäjä kirjoittaa hakusanakenttään. Ontologioihin perustuu myös ns. moninäkymähaku (Hyvönen ym. 2004), joka niin ikään helpottaa tiedonhakua. Moninäkymähaussa käyttäjä voi katsella ontologisesti kuvattua tietoa useasta eri näkökulmasta ja näkymien välillä siirtyessään etsiä tietoa tuntematta tarkemmin sen kuvailussa käytettyä sanastoa. Tällaisessa tiedonhakutavassa käyttäjä tutustuu tietoa hakiessaan automaattisesti myös hakemaansa tietoa lähellä oleviin käsitteisiin ja saa niiden kautta mahdollisesti sellaista lisätietoa, jota perinteiset tiedonhakumenetelmät eivät olisi tarjonneet.

2.3 Resurssit ja niihin viittaaminen

Semanttisen webin kehittämisessä yhtenä päämääränä on tarjota monipuolisia työvälineitä erilaisten resurssien kuvailuun ja niiden välisten suhteiden ilmaisemiseen. Pohjana tällaiselle ilmaisulle toimivat XML- ja RDF-kielet, joissa erilaisiin asioihin viitataan URI:lla (Berners-Lee ym. 2001). URI voi olla URL (Universal Resource Locator), eli web-osoite, tai jonkinlainen muu yksikäsitteinen tunniste. URI:n kautta ei välttämättä tarvitse päästä käsiksi resurssiin; URI-skeemoja on määritelty web-osoitteiden (URL) lisäksi myös muille objekteille, kuten puhelin- ja ISBN-numeroille (International Standard Book Number) sekä maantieteellisille paikoille (Antoniou & Harmelen, 2004, s.64).

URI:lla identifioitavat asiat voidaan Boothin (2003) mukaan jakaa kahteen ryhmään. Ensimmäiseen kuuluvat ne asiat, jotka sijaitsevat webissä. Tällaisia asioita ovat esimerkiksi www-sivut ja muut vastaavat dokumentit, joihin tyypillisesti voidaan osoittaa URL:n avulla. Toisessa ryhmässä ovat taas kaikki webin ulkopuolella olevat asiat, jotka voidaan jakaa edelleen kahteen kategoriaan: fyysisiin objekteihin (esim. autot, ihmiset, rakennukset jne.) ja abstrakteihin käsitteisiin (koko, väri, rakkaus jne.). Esimerkiksi www-sivua kuvailevassa annotaatiossa viitataan webissä olevaan resurssiin, koska sivu voidaan hakea sille määritellyn URL:n perusteella. Webin ulkopuolella olevaan resurssiin viitataan taas esimerkiksi jostain henkilöstä kertovassa annotaatiossa. Tässä tapauksessa ei välttämättä ole olemassa www-sivua tai muuta dokumenttia, joka jollain tavalla *edustaisi* kyseistä henkilöä webissä, tai kertoisi mahdollisesti jotain lisätietoa hänestä. Henkilöön viittaavalla URI:lla ei voida toisin sanoen hakea samalla tavalla resurssia, kuin webissä olevaan dokumenttiin viittaavalla URI:lla voidaan. Sen sijaan URI toimii henkilön tunnisteena, jonka avulla voidaan eri yhteyksissä yksikäsitteisesti viitata juuri tiettyyn henkilöön.

Semanttisen webin resurssien tunnistamiseen liittyvistä asioista on käyty paljon keskustelua (Booth, 2003, Hawke, 2001), eikä vielä ole löydetty esimerkiksi selkeää vastausta kysymykseen, mikä olisi sopivin tapa tunnistaa ihmisiä (Antoniou & van

Harmelen, 2004). Tässä työssä ei käsitellä syvemmin resurssien tunnistamiseen liittyviä kysymyksiä, mutta annotaatioiden luomisprosessin kannalta on kuitenkin olennaista huomioida, että annotaatioissa voidaan URI-skeemoja käyttäen viitata fyysisiin objekteihin, abstrakteihin käsitteisiin sekä web-dokumentteihin.

3 ANNOTOINTI SEMANTTISESSA WEBISSÄ

Tässä luvussa käsitellään semanttisen webin annotointia. Luvussa kerrotaan minkä tyyppisiä annotaatioita webin yhteydessä esiintyy ja minkälaisilla tavoilla niitä voidaan tuottaa. Tämän lisäksi käsitellään annotaatioiden liittämistä dokumentteihin ja esitellään lyhyesti annotointiin liittyviä automaattisia menetelmiä.

3.1 Annotaatio

Annotaatiolla on perinteisesti viitattu tekstiin liitettyyn kommenttiin tai huomautukseen, jonka tekijä on laatinut omaksi muistiinpanokseen tai mahdollisesti tiedoksi muille lukijoille. Semanttisen webin myötä annotaation merkitys dokumenttia kuvaavana merkintänä on laajentunut, kun sillä on ryhdytty viittaamaan webin resursseja tietokoneille kuvailevaan metatietoon. Kahan ym. (2001) määritelmän mukaan annotaatio on webin yhteydessä URI:lla identifioitu, dokumenttiin liitetty kommentti, joka on joko dokumentin kirjoittajan tai jonkin muun osapuolen laatima. Yleisemmällä tasolla he määrittelevät annotaatiot metatiedoksi, jotka liittävät huomautuksia olemassa oleviin dokumentteihin.

Ontologiaperustaisessa annotoinnissa[6] annotaatio ei välttämättä sisällä ainoastaan dokumenttiin liitettyä tekstimuotoista kommenttia, vaan voi muodostua monipuolisemmasta dokumenttia kuvailevasta tiedosta, joka voi sisältää viitteitä muihin annotaatioihin, webin resursseihin sekä erilaisiin ontologioissa määriteltyihin käsitteisiin. Ontologioihin pohjautuvan annotoinnin yhteydessä annotaatiolla saatetaan viitata metatietoon, joka kuvailee mitä tahansa webissä tai sen ulkopuolella olevaa resurssia. Annotaatio voi toisin sanoen kuvailla dokumenttien lisäksi myös muita asioita, kuten esimerkiksi ihmisiä ja paikkoja. Koska annotointia tarkastellaan usein erilaisten dokumenttien näkökulmasta, on monissa tapauksissa kuitenkin selkeää kutsua annotaatioiksi juuri dokumentteja kuvaavaa metatietoa.

3.2 Annotaatiotyypit

3.2.1 Tekstiannotaatiot

Marshallin (1998) mukaan annotaatioissa ilmaistavan tiedon kuvailutapa voi vaihdella formaalista epäformaaliin. Formaaleimpia ovat annotaatiot, joissa tieto on kuvattu yhteisesti sovittuja standardeja ja nimeämiskäytäntöjä soveltaen. Epäformaaleja annotaatioita ovat puolestaan esimerkiksi kirjan marginaaliin kirjoitetut huomautukset tai muistiinpanot. Webissä olevat annotaatiot voidaan jakaa kahteen päätyyppiin sen mukaan, ovatko ne tarkoitettu ensisijaisesti ihmisen vai tietokoneen tulkittavaksi. Ihmisille tarkoitettuja, luonnollista kieltä sisältäviä annotaatioita voidaan kutsua *tekstiannotaatioiksi*. Ne ovat tyypillisesti dokumenttiin jollain sopivalla tavalla liitettyjä epäformaalisti kuvattuja merkintöjä, joiden tarkoitus on täydentää dokumentin sisältöä ja joita lukija voi

[6] *Annotoinnilla* viitataan tapahtumaan, jossa annotaatioita, eli metatietoa tuotetaan

dokumenttia lukiessaan tarkastella. Tällaisten annotaatioiden muodostamista tuetaan esimerkiksi Annotea-järjestelmässä (Kahan ym. 2001), jossa käyttäjä voi tarkoitusta varten suunnitellulla web-selaimella katsella www-sivuja ja niihin liitettyjä annotaatioita.

Tekstiannotaatiot saattavat tarjota lukijalleen tärkeää ja hyödyllistä lisätietoa dokumentista, mutta tietokoneiden näkökulmasta ne ovat vain uusia dokumenttiin liitettyjä monimerkityksellisiä sanoja, joista myös varsinainen dokumentti muodostuu. Koska tietokoneet eivät pysty tulkitsemaan tällaisten sanojen merkityksiä, pyritään semanttisessa webissä muodostamaan annotaatiot formaalilla tavalla ontologisesti kuvailtuja käsitteistöjä käyttäen.

3.2.2 Ontologiaperustaiset annotaatiot

Ontologiaperustaisten annotaatioiden tarkoituksena on kuvailla webissä olevia dokumentteja siten, että ihmisten lisäksi myös tietokoneet pystyisivät tulkitsemaan niitä. Tekstiannotaatioiden tapaan ne muodostavat dokumentteihin liittyvän lisätietokerroksen, mutta tässä tapauksessa tieto on kohdistettu ihmisten sijasta ensisijaisesti tietokoneille. Ontologioihin pohjautuvat annotaatiot ovat tärkeässä asemassa, kun pyritään kehittämään tietojärjestelmiä, jotka pystyvät toimimaan semanttisesti yhteensopivalla tavalla (Hyvönen ym. 2005). Ontologisesti kuvaillun tiedon avulla voidaan esimerkiksi tiedonhaussa saavuttaa suurempi tarkkuus verrattuna siihen, että haettavaa tietoa olisi kuvailtu ainoastaan luonnollisella kielellä.

Handschuh ja Staab (2003) käyttävät ontologiaperustaisista annotaatioista nimitystä *suhteellinen metadata* (relational metadata). Heidän mukaansa suhteellinen metadata muodostuu ontologioissa määriteltyjen luokkien ja ominaisuuksien ilmentymistä sekä niiden välisistä suhteista. Kiryakovin ym. (2003) mukaan ontologioihin pohjautuvat annotaatiot yhdistävät tekstissä olevat entiteetit[7] niitä vastaaviin semanttisiin kuvauksiin.

Ontologioita käytetään annotoinnissa tyypillisesti siten, että annotaatiot muodostetaan ontologioissa kuvattujen luokkien sekä niille määriteltyjen ominaisuuksien ilmentymistä ja ilmaistaan jollain sopivalla tavalla niiden liittyminen kuvailtaviin resursseihin. Annotaatioiden ja dokumenttien välinen suhde voidaan määrittää liittämällä annotaatiot osaksi dokumenttia[8], tai sisällyttämällä annotaatioon dokumentin URL-osoite. Handschuh ja Staab (2003) jakavat ontologiaperustaiset annotaatiot ontologian luokkien ilmentymiin, sekä niihin liittyviin teksti- ja objektiarvoisiin ominaisuuksiin. Tekstiarvoisilla ominaisuuksilla viitataan niihin ominaisuuksiin, jotka saavat arvokseen luonnollisen kielen sanoja. Nämä muistuttavat edellä kuvattuja tekstiannotaatioita ja ovat tarpeellisia esimerkiksi silloin, kun jostain ilmentymästä esitetään ihmisen tulkittavaksi tarkoitettua tietoa. Tällaista tietoa voi olla esimerkiksi henkilöä kuvaavassa annotaatiossa henkilön

---

[7] *Entiteetillä* viitataan tässä johonkin olemassa olevaan asiaan, kuten ihmiseen. Entiteettiin voidaan viitata siihen liitetyllä nimellä, jolloin voidaan puhua myös *nimetystä entiteetistä*. Tällaisia ovat esimerkiksi ”Alvar Aalto”, ”Suomi” ja ”Teknillinen korkeakoulu”.

[8] Käytännössä liittämisellä tarkoitetaan sitä, että annotaatio kirjoitetaan johonkin kohtaan dokumenttia.

nimi. Tekstiarvoisia ominaisuuksia voidaan kutsua myös *literaaliominaisuuksiksi* ja niiden saamia arvoja *literaaliarvoiksi*. Objektiarvoiset ominaisuudet sisältävät puolestaan formaalisti kuvattua tietoa erilaisten resurssien välisistä suhteista ja näin ollen muodostavat semanttisen webin kannalta tärkeimmän metatiedon. Tällaiset ominaisuudet saavat arvokseen esimerkiksi ontologioissa määriteltyjä luokkia tai niiden ilmentymiä. Objektiarvoisia ominaisuuksia voidaan kutsua myös *objektiominaisuuksiksi* ja niiden saamia arvoja *objekti-* tai *ilmentymäarvoiksi*. Objektiominaisuus voi saada arvokseen esimerkiksi johonkin webin ulkopuoliseen resurssiin, kuten henkilöön viittaavan ilmentymän. Käyttämällä tällaisessa tapauksessa ilmentymää, voidaan osoittaa yksikäsitteisesti juuri tiettyyn henkilöön. Tämä ei olisi mahdollista käyttämällä pelkästään henkilön nimeä, koska useammalla henkilöllä voi olla sama nimi, eikä koneellisesti tehtävällä päättelyllä pystytä erottelemaan mistä henkilöstä tällöin on kyse. Ilmentymien lisäksi objektiominaisuus voi saada arvokseen myös jonkin ontologian resurssin. Tässäkin tapauksessa ontologian avulla voidaan yksikäsitteisesti viitata esimerkiksi juuri tiettyyn käsitteeseen ja välttää näin sanojen monimerkityksellisyyteen liittyviä tulkintaongelmia, joihin joudutaan kun arvona käytetään luonnollisen kielen sanoja.

3.3 Annotointimenetelmät

3.3.1 Menetelmien jaottelu

Schreiber ym. (2001) jaottelevat ontologiaperustaiseen annotointiin liittyvät ontologiat kahteen eri ryhmään:

- Annotaatio-ontologiat (annotaatioskeemat), jotka määrittelevät kuvattavasta aineistosta riippumattomasti annotaation rakenteen.
- Aihekohtaiset sanastot (subject matter vocabulary), jotka tarjoavat tiettyyn aihepiiriin liittyen annotaatioissa käytettävän sanaston.

Kun annotoinnissa hyödynnetään annotaatio-ontologioita, voidaan annotointia kutsua *skeemaperustaiseksi*. Siinä annotaatio-ontologia kuvaa annotaatioiden rakenteen määrittelemällä mallin sille, miten aihekohtaisessa sanastossa määriteltyjä käsitteitä liitetään annotoitavaan resurssiin. Annotointia on mahdollista suorittaa myös siten, että annotoinnissa käytetään ainoastaan aihekohtaisia sanastoja. Tällaista annotointia voidaan kutsua *vapaaksi annotoinniksi*, koska siinä sanastojen käyttöä ei ohjata annotaatio-ontologian avulla. Seuraavissa luvuissa on kerrottu tarkemmin kummastakin menetelmästä.

3.3.2 Vapaa annotointi

Ontologioihin tai muihin vastaaviin sanastoihin pohjautuvassa vapaassa annotoinnissa dokumentteja kuvaillaan siten, että niihin liitetään dokumenttien sisältöä jollain halutulla tavalla kuvaavia ontologisia käsitteitä. Vapaassa annotoinnissa käytettävät ontologiat

voivat kuvata jossain tietyssä aihepiirissä käytettävän sanaston, tai olla YSO:n [9] (Hyvönen ym. 2005) kaltaisia yleissanastoja. Sanasto-ontologioita voidaan kutsua niiden käyttötarkoituksen vuoksi myös *referenssiontologioiksi*. Vapaata annotointia on toteutettu mm. SemTag-projektissa (Dill ym. 2003), jossa suurta joukkoa www-sivuja annotoitiin automaattisesti yhdistämällä dokumenteissa olevia entiteettejä laajaan TAP-tietämyskantaan[10]. SemTag:ssä annotaatioina toimivat HTML-tiedostossa alkuperäiseen tekstin lomaan upotetut tietämyskantaan osoittavat viittaukset. Esimerkiksi lause ”The Chicago Bulls announced yesterday that Michael Jordan will...” annotoitiin SemTag:llä siten, että lause esitettiin annotoinnin jälkeen muodossa:

```
"The <resource ref="http://tap.stanford.edu/BasketballTeam_Bulls">Chicago
Bulls</resource>announced yesterday that <resource
ref="http://tap.stanford.edu/AthleteJordan,_Michael">Michael
Jordan</resource> will... "
```

(annotaatiot on korostettu punaisella tekstillä)

SemTag:in tuottamassa annotaatiossa

```
<resource ref="http://tap.stanford.edu/BasketballTeam_Bulls">Chicago
Bulls</resource>
```

ilmaistaan, että merkkijono “Chicago Bulls” viittaa resurssiin, jonka URI on `http://tap.stanford.edu/BasketballTeam_Bulls`. URI:n tehtävänä on ilmaista, että kyseessä on juuri tietty ”Chigago Bulls”-niminen koripallojoukkue ja mahdollisesti osoittaa paikkaan, joka tarjoaa tarkempaa lisätietoa resurssista.

Vapaa annotointi eroaa skeemaperustaisesta annotoinnista siten, että siinä ei käytetä skeemaa, joka määrittelisi dokumentti- tai resurssikohtaisesti annotaatioiden rakenteen ja sitä kautta asiat, jotka annotoitavasta resurssista tulee kuvailla. Vapaassa annotoinnissa ei myöskään pystytä skeemaperustaisen annotoinnin tapaan ilmaisemaan *millä tavalla* annotaatiot liittyvät kuvailtavaan resurssiin. Koska annotaatioiden rakennetta ei vapaassa annotoinnissa ole yksikäsitteisesti kuvattu, samasta dokumentista on mahdollista luoda monia erilaisia annotaatioita, vaikka annotaatioiden kuvailuun käytettäisiinkin samaa kieltä ja ontologiaa (Naing ym. 2002). Tällaisessa tapauksessa annotaatiot muodostuvat satunnaisten luokkien ja ominaisuuksien instansseista ja niiden merkitys saattaa helposti jäädä epäselväksi (Handschuh & Staab, 2002). Koska annotaatioiden rakenne ei ole yhtenevä, ei vapaasti annotoidulle aineistolle ole välttämättä mahdollista toteuttaa esimerkiksi näkymäpohjaista hakua (Hyvönen ym. 2004), joka vaatii sitä, että annotoitavan aineiston jokaisesta osasta kuvaillaan tietyt yhteiset piirteet.

---

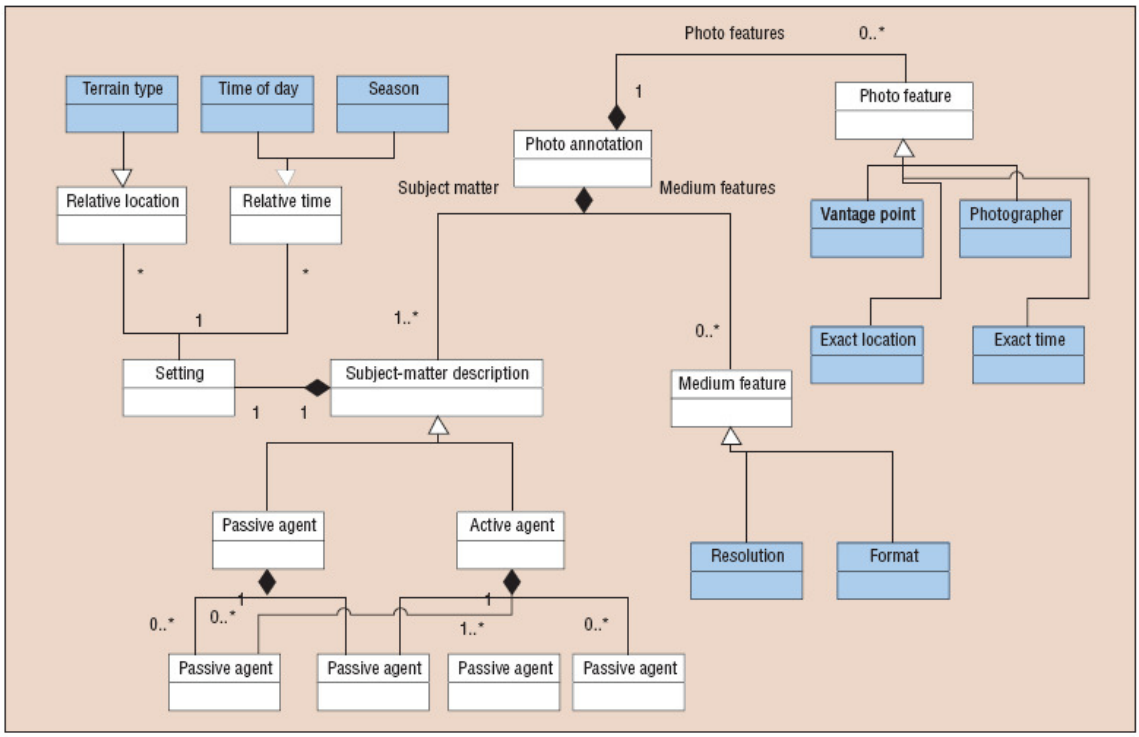
[9] Yleinen Suomalainen Ontologia

[10] http://tap.stanford.edu/

3.3.3 Skeemaperustainen annotointi

Annotaatioiden muodostamista on yleensä tarkoituksenmukaista ohjata määrittelemällä yksikäsitteisesti asiat, joita annotoitavasta aineistosta halutaan kuvailla. Tällaista määrittelyä kutsutaan annotaatioskeemaksi tai -ontologiaksi ja se määrittelee tavan, jolla annotaatioilmentymät muodostetaan (Naing ym. 2002). Annotaatiot muodostuvat annotaatioskeemana toimivan ontologian luokkien ja ominaisuuksien ilmentymistä, joihin liitetään viittaus kuvailtuun resurssiin. Annotaatioskeemoja voidaan verrata tietokantanäkymiin, joiden tarkoituksena on peittää alla olevien tietokantojen monimutkaisuus käyttäjältä, joka syöttää ja hakee niistä tietoa (Kettler ym. 2005). Annotaatioskeemojen avulla annotoijan ei tarvitse muodostaa annotaatioita laajojen referenssiontologioiden, kuten YSO:n pohjalta, vaan voi tuottaa niitä aihepiiriltään rajatumpia ja rakenteeltaan yksinkertaisempia ontologioita käyttäen. Annotaatioskeeman avulla annotointiprosessia voidaan ohjata erilaisilla rajoituksilla, joiden avulla voidaan määritellä esimerkiksi eri ominaisuuksille sallittuja arvoalueita tai ontologioita, joista arvot tulee hakea (Geurts ym. 2005). Geneerinen annotointisovellus voi hyödyntää annotaatioskeemaa käyttöliittymänsä muodostamisessa, jolloin skeema toimii mallina käyttöliittymään luotavalle lomakkeelle, jonka kautta annotaatiot syötetään.

Schreiber ym. (2001) kuvailevat valokuvien annotointimenetelmän, jossa valokuvien annotaatiot muodostetaan tarkoitusta varten kehitetyn annotaatioskeeman perusteella. Skeemassa on määritelty valokuvien sisältöjen kuvailussa käytettävät ominaisuudet, joiden arvot haetaan valokuvien aihepiiriin liittyvästä referenssiontologiasta. Annotaatioskeeman luokkarakenne on esitetty UML-kaaviona kuvassa 2.



Kuva 2. *Annotaatioskeeman UML-kaavio (Schreiber ym. 2001)*

Tarkastelemalla kuvaa 2 nähdään, miten annotaation rakenne määritellään skeemassa. Valokuvan annotaatio (Photo annotation) koostuu ainakin yhdestä asiasisällön kuvauksesta (Subject-matter description) ja valinnaisesta määrästä valokuvan ominaisuuksia (Photo feature) sekä tallennusominaisuuksia (Medium feature). Asiasisällön kuvauksella on sisäinen rakenne, joka edelleen muodostuu kuvissa esiintyvistä toimijoista (Agent) ja tapahtumaympäristöstä (Setting).

Skeemaperustaista annotointia sovelletaan mm. Annotea-projektissa (Kahan ym. 2001), jossa on määritelty ympäristö www-sivujen annotointiin ja annotaatioiden hyödyntämiseen[11]. Annoteassa www-sivujen lukijat voivat annotoida sivuja ja tallentaa annotaatiot keskitetylle annotaatiopalvelimelle. Annotaatioita voidaan katsella Amaya-selaimella[12], joka on yhteydessä annotaatiopalvelimeen ja näyttää selaimeen avatun www-sivun annotaatiot, mikäli se löytää niitä palvelimelta. Annoteassa annotaatiot muodostetaan RDF:lla kuvatun annotaatioskeeman pohjalta[13]. Skeemassa on määritelty yleinen annotaatioiden yläluokka ”Annotation”, joka kuvailee erilaisten annotaatioiden yhteiset perusominaisuudet, jotka on lueteltu taulukossa 1. Annotation-luokan ominaisuudet kuvaavat pääasiassa itse annotaatiota, varsinaisen resurssin kuvailuun on määritelty yksi ominaisuus. Annotea-järjestelmässä tarkoituksena on, että erilaiset annotaatioita tuottavat käyttäjäryhmät luovat Annotation-luokalle uusia aliluokkia, joille määriteltävien lisäominaisuuksien avulla voidaan kuvailla tarkemmin johonkin tiettyyn aihepiiriin liittyviä dokumentteja. Annotealla suoritettu annotointi ei ole ontologioihin pohjautuvaa, eikä järjestelmässä oteta kantaa siihen, minkä tyyppisiä arvoja annotaatioiden ominaisuuksille tulisi määritellä.

Taulukko 1. *Annotea-järjestelmän Annotation-luokan ominaisuudet*

Ominaisuus	**Kuvaus**
`annotates`	Annotoitava resurssi
`author`	Annotaation luojan tai luojaorganisaation nimi
`body`	Annotaation sisältö
`context`	Paikka, johon annotaatio pääasiassa annotoitavassa resurssissa viittaa
`created`	Annotaation luontipäivämäärä
`modified`	Päivämäärä, jolloin annotaatiota on viimeksi muokattu
`related`	Annotaation suhde muihin resursseihin

Dublin Core[14] on yksinkertainen metatietoskeema, jota käytetään yleisesti webissä olevien dokumenttien kuvailuun. Siinä on määritelty joukko dokumenttia yleisesti kuvailevia ominaisuuksia, kuten *tekijä*, *kuvaus*, *formaatti*, *päivämäärä* jne. RDF-pohjaisia annotaatioita varten Dublin Core määrittelee RDF-skeeman, joka sisältää 15 tällaista

[11] http://www.w3.org/2001/Annotea/

[12] http://www.w3.org/Amaya/

[13] Annotean annotaatioskeema löytyy osoitteesta: http://www.w3.org/2000/10/annotation-ns#

[14] http://www.dublincore.org

ominaisuutta. Dublin Coren skeema ei itsessään ole riittävä semantiikaltaan rikkaampien annotaatioiden tuottamisessa, koska siinä ei esimerkiksi määritellä formaalisti ominaisuuksien tyyppejä eikä niiden arvoalueita. Dublin Coren tarjoama sanasto on kuitenkin yleisyytensä vuoksi usein käytössä osana muita, mahdollisesti semanttisesti rikkaammin kuvattuja annotaatioskeemoja. Esimerkiksi Annotean annotaatioskeemassa oleva ominaisuus ”author” on määritelty Dublin Coren ominaisuuden ”creator” aliominaisuudeksi.

Annotaatioskeema voidaan suunnitellaan siten, että osaa sen luokista käytetään dokumenttien annotointiin ja osaa taas muiden resurssien, kuten esimerkiksi henkilöiden, paikkojen sekä erilaisten abstraktien käsitteiden kuvailuun. Tällöin skeeman luokat voidaan jakaa niiden käyttötarkoituksen mukaan kahteen eri ryhmään: *annotaatioluokkiin* ja *referenssiluokkiin*. Tällaisessa tapauksessa voidaan myös määritellä, että annotaatioiksi kutsutaan dokumentteja kuvaavien annotaatioluokkien ilmentymiä. Referenssiluokkien ilmentymät puolestaan muodostavat ontologisesti kuvatun *tietämyskannan*, jota käytetään määriteltäessä arvoja annotaatioluokkien objektiominaisuuksille. On huomattava, että edellä kuvattu erottelu pohjautuu ainoastaan luokkien merkitykseen annotoinnissa. Syntaktisesti annotaatio- ja referenssiluokat sekä niiden ilmentymät eivät eroa toisistaan. Kun annotaatioskeema suunnitellaan edellä kuvatulla tavalla, on sillä kaksi tehtävää: se määrittelee annotaatioluokkien avulla annotaatioiden rakenteen ja toisaalta toimii referenssiluokkiensa kautta tietämyskantana, jota voidaan hyödyntää uusia annotaatioita muodostettaessa. Annotaatioskeema voidaan toteuttaa myös niin, että kaikkia sen luokkia voidaan käyttää minkä tahansa resurssin kuvailuun. Tällöin edellä esitetty skeeman luokkien jaottelu ei ole mielekästä, koska luokilla ei ole selkeää roolijakoa. Vastaavasti skeema voidaan suunnitella siten, että siinä määriteltyjen luokkien ilmentymät kiinnitetään aina johonkin dokumenttiin. Tällaisessa tapauksessa skeeman ainoana tehtävänä on määrittää dokumentteja kuvailevien annotaatioiden rakenne (vrt. kuvan 2 skeema). Annotaatioiden objektiominaisuuksien arvot voidaan tällaisessa tapauksessa hakea esimerkiksi jostain ulkoisesta referenssiontologiasta.

Eräs ontologioihin ja siten myös annotaatioskeemoihin liittyvä ongelma on ratkaista erilaisten ontologioiden semanttinen yhteentoimivuus. Mikäli eri aineistoja annotoidaan erilaisia ontologioita (skeemoja) käyttäen, eivät annotaatiot ole yhteensopivia ilman erillistä ontologioiden välistä kuvausta (*ontology mapping*). Siinä ontologiat yhdistetään käsitteellisellä tasolla semanttisesti toisiinsa ja muunnetaan lähdeontologian käsitteet kohdeontologian käsitteiksi kuvauksessa muodostettuja suhteita käyttämällä (Silva & Rocha, 2003). Skeemojen välisten kuvauksien lisäksi annotaatioiden semanttista yhteentoimivuutta voidaan parantaa käyttämällä annotoinnissa eri skeemoille yhteisiä tietämyskantoja ja referenssiontologioita.

3.3.4 Menetelmien vertailua

Vapaan ja skeemaperustaisen annotoinnin merkittävin ero syntyy siitä, miten niiden yhteydessä hyödynnettävät ontologiat on suunniteltu ja miten ontologioita käytetään annotoinnissa. Skeemaperustaisessa annotoinnissa skeeman annotaatioluokat muodostetaan

usein siten, että ne sisältävät joukon dokumenttia ja sen sisältöä kuvaavia ominaisuuksia, joiden avulla pystytään kuvailemaan kokonainen dokumentti. Vapaassa annotoinnissa yhden dokumentin kuvaus muodostuu taas usein useammasta annotaatiosta. Tässä tapauksessa annotaatiot saattavat olla ainoastaan annotoituun dokumenttiin tai mahdollisesti johonkin sen tarkempiin rakenneosiin liitettyjä annotaatioluokkien URI-tunnisteita, kuten luvussa 3.3.2 esitettiin.

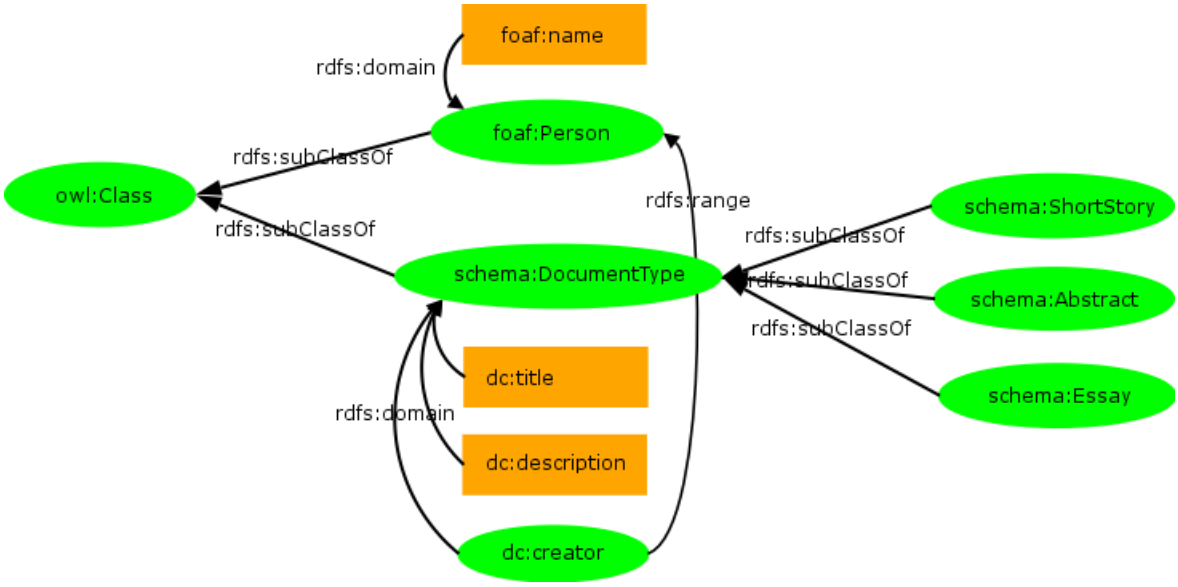
Semanttista webiä varten kehitetyt annotointimenetelmät ja -järjestelmät voidaan karkeasti jakaa sen perusteella, kumpaa edellä esitetyistä annotointimenetelmistä niissä hyödynnetään. Annotaatioskeemojen voidaan nähdä soveltuvan hyvin mm. sellaisiin tilanteisiin, joissa etukäteen tiedetään suhteellisen tarkasti mitä aineistosta halutaan annotaatioilla kertoa ja minkälaisia dokumentteja tullaan kuvailemaan. Esimerkiksi MuseoSuomi-projektissa (Hyvönen ym. 2003) kuvailtiin suomalaisten museoiden kokoelmissa olevia museoesineitä tarkoitusta varten laaditun skeeman mukaisesti ja näin kaikista esineistä saatiin koottua yhtenäiset perustiedot. Koska kaikilla museoesineillä oli yhteisiä piirteitä, kuten esimerkiksi valmistuspaikka ja -materiaali, oli niitä selkeää kuvailla yhteisellä annotaatioskeemalla. Mikäli annotoidaan hyvin heterogeenistä aineistoa, saattaa annotaatioskeeman suunnittelu olla haasteellista, koska etukäteen ei tarkasti tunneta kuvailtavan aineiston sisältöä ja rakennetta. Tällöin ratkaisuna voi olla mahdollisimman yleisen annotaatioskeeman toteuttaminen tai vapaan annotoinnin soveltaminen. Vapaan annotoinnin käyttö voi olla perusteltua myös silloin, kun annotaatioita tuotetaan automaattisesti, eikä monimutkaisempien semanttisten suhteiden tunnistaminen ole sen vuoksi mahdollista.

3.4 Annotaatioiden liittäminen dokumentteihin

Semanttisessa webissä annotaatiot voidaan liittää joko osaksi dokumentteja tai tallentaa erillään niistä, kuten esimerkiksi Annoteassa (Kahan ym. 2001). Annotaatioiden erottaminen dokumenteista on perusteltua mm. siksi, että erikseen tallennettuna niiden ylläpito on helpompaa ja tiettyyn dokumenttiin voidaan liittää useampia erilaisia käyttäjäkohtaisia annotaatioita (Kiriyakov ym. 2003). Erottamista tukee myös se, että annotoijalla ei usein ole edes mahdollisuutta kirjoittaa annotaatiota dokumenttiin. Tämän voi estää esimerkiksi dokumentin formaatti sekä usein se, että dokumentin omistaa jokin toinen taho, minkä johdosta annotoijalla ei ole kirjoitusoikeutta siihen.

Kun annotaatio tallennetaan erillään annotoidusta dokumentista, tarvitaan keino viitata annotaatiosta dokumenttiin. RDF-pohjaisessa annotoinnissa tällaiseen viittaamiseen voidaan käyttää kahta eri tapaa. Annotaatio voidaan liittää dokumenttiin ilmaisemalla dokumentin olevan annotaatiossa käytetyn luokan ilmentymä. Tällaista viittaustapaa voidaan kutsua dokumenttien luokitteluksi, koska siinä dokumentti sidotaan vahvasti tiettyyn luokkaan ilmaisemalla sen olevan kyseisen luokan ilmentymä. Toisena vaihtoehtona on liittää annotaatio dokumenttiin jonkin ominaisuuden avulla, kuten Annotean annotaatioskeemassa tehdään ominaisuudella ”annotates”. Tällöin ominaisuuden arvoksi tulee annotoidun dokumentin URL-osoite.

Dokumenttien luokittelussa annotoinnin voidaan ajatella olevan kuvailtavien dokumenttien jaottelua erilaisiin luokkiin, jossa dokumentin luokan määrittelee sitä kuvaavassa annotaatiossa käytetty annotaatioluokka. Tällaisessa annotoinnissa annotaatioluokat on tarkoituksenmukaista nimetä siten, että nimet vastaavat haluttua luokittelua. Esimerkiksi erityyppisiä kirjoituksia kuvaava annotaatioskeema voisi sisältää annotaatioluokat ”tiivistelmä”, ”essee”, ”novelli”, ”artikkeli” jne. Skeeman pohjalta muodostettavissa annotaatioissa ilmaistaan tällöin, että jokainen annotoitu dokumentti on ilmentymä jostain näistä luokista. Dokumenttien luokittelua soveltavaa annotointia on havainnollistettu kuvassa 3. Siinä on esitetty RDF-verkko, jossa kuvaillaan yksinkertainen annotaatioskeema. Skeemassa on määritelty annotaatioluokat `schema:DocumentType` (dokumenttityyppi) ja sen aliluokat `schema:Abstract` (tiivistelmä), `schema:Essay` (essee) ja `schema:ShortStory` (novelli) sekä referenssiluokka `foaf:Person`. Skeeman RDF/XML-kuvaus on liitteessä 1.



Kuva 3. *Annotaatioskeema*

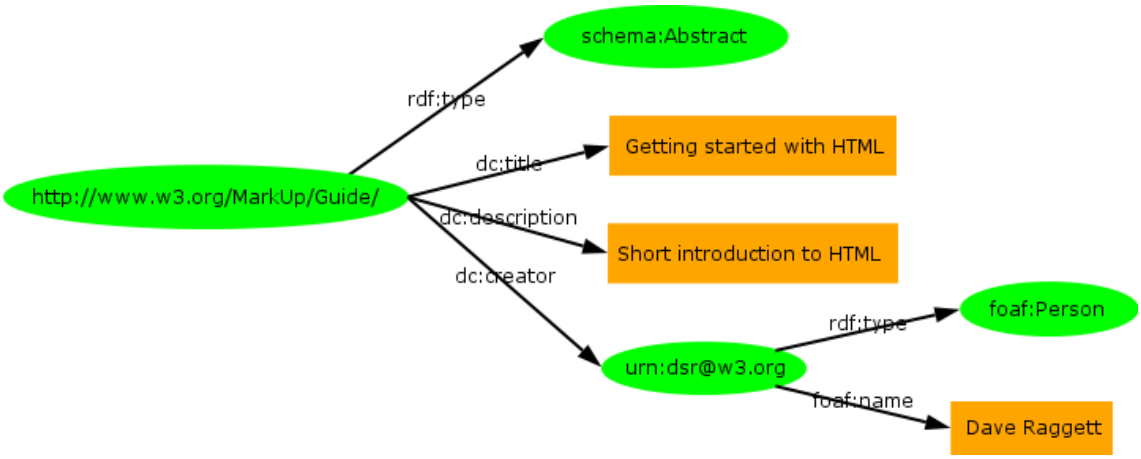
`schema:DocumentType`-luokan aliluokat ovat erilaisia dokumenttityyppejä kuvailevia annotaatioluokkia ja suunniteltu dokumenttien luokitteluun. `foaf:Person`-luokka on taas referenssiluokka, jonka ilmentymiin voidaan viitata annotaatioluokkien ominaisuudesta `dc:creator`. Seuraavassa on RDF/XML:lla kuvattu esimerkki annotaatioskeeman pohjalta luodusta annotaatiosta, jossa on kuvattu www-sivu `http://www.w3.org/MarkUp/Guide/`, jonka tekijä on henkilö ”Dave Raggett”:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:owl="http://www.w3.org/2002/07/owl#"
         xmlns:schema="http://www.seco.hut.fi/annotation#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/"
         xmlns:dc="http://purl.org/dc/elements/1.1/">

  <schema:Abstract rdf:about="http://www.w3.org/MarkUp/Guide/">
     <dc:title>Getting started with HTML</dc:title>
```

```
      <dc:description>Short introduction to HTML</schema:description>
      <dc:creator rdf:resource="urn:dsr@w3.org"/>
  </schema:Abstract>

  <foaf:Person about="urn:dsr@w3.org">
    <foaf:name>Dave Raggett</foaf:name>
  </rdf:Description>
</rdf:RDF>
```

(annotaatio on korostettu punaisella, varsinaisen skeeman määrittely on selkeyden vuoksi jätetty pois)

Annotaatiossa ilmaistaan, että dokumentti jonka URL on `http://www.w3.org/MarkUp/Guide/` on luokan `schema:Abstract` (tiivistelmä) ilmentymä. Tällainen annotaatio ilmaisee toisin sanoen dokumentin kuuluvan luokan ”tiivistelmä” määrittämien yksilöiden joukkoon. Edellä kuvattu annotaatio on esitetty RDF-verkkona kuvassa 4.



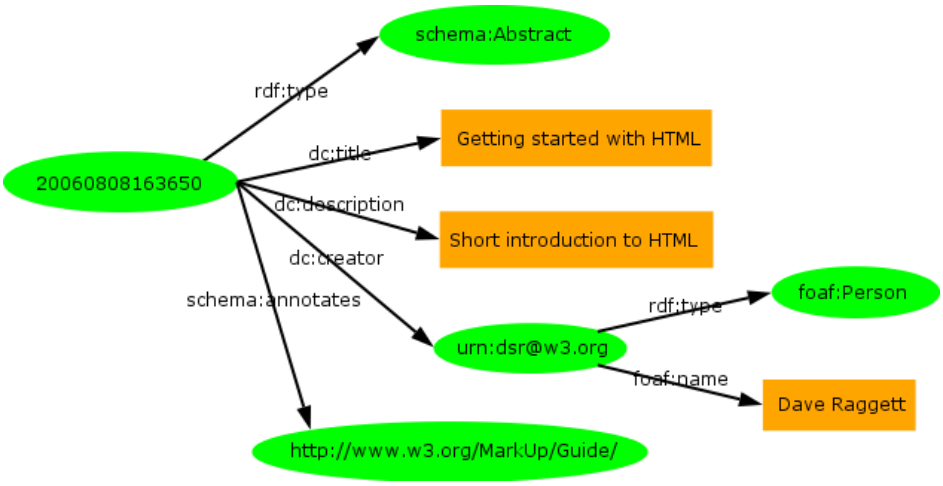
Kuva 4. *Annotaatioskeeman pohjalta muodostettu annotaatio*

Esitettyä annotaatioesimerkkiä tarkastelemalla voidaan havaita miten annotaation rakenteen kuvailevan annotaatioluokan `schema:Abstract` sekä referenssiluokan `foaf:Person` roolit eroavat annotaatiossa. `foaf:Person`-luokan ilmentymä on määritelty yhdeksi annotaatioluokan ominaisuuksista ja se toimii näin osana varsinaista annotaatiota, jonka `schema:Abstract`-luokan ilmentymä muodostaa. `foaf:Person`-luokan ilmentymää voitaisiin käyttää arvona myös muissa annotaatioissa.

Annotaation ja dokumentin välinen yhteys voidaan kuvailla edellä esitetyn dokumenttien luokittelun lisäksi myös jonkin annotaatioskeemassa määritellyn ominaisuuden avulla. Tällainen viittaustapa muistuttaa Annoteassa käytettävää ”annotates”-ominaisuutta. Sitä voidaan hyödyntää esimerkiksi silloin, kun ei haluta ilmaista dokumentin olevan minkään luokan ilmentymä tai kun halutaan tarkemmin kuvata dokumentin ja annotaation välistä suhdetta. Kuvassa 4 esitetty web-sivun annotaatio voitaisiin liitää dokumenttiin ominaisuuden kautta esimerkiksi seuraavalla tavalla:

```
<schema:Abstract rdf:ID="20060808163650">
  <schema:annotates rdf:resource="http://www.w3.org/MarkUp/Guide/"/>
  <dc:title>Getting started with HTML</dc:title>
  <dc:description>Short introduction to HTML</schema:description>
  <dc:creator rdf:resource="urn:dsr@w3.org"/>
</schema:Abstract>
```

Tässä tapauksessa annotaatiossa kerrotaan, että luokka `schema:Abstract` liittyy dokumenttiin `http://www.w3.org/MarkUp/Guide/`, mutta ei ilmaista dokumentin olevan `schema:Abstract`-luokan ilmentymä. Annotaation tunnisteena käytetään dokumentin URL:n sijaan annotaation yksilöivää, geneerisesti luotua tunnistetta `20060808163650`. Annotaation ja dokumentin välinen yhteys on ilmaistu ominaisuudella `schema:annotates`, jonka arvona on dokumentin URL. Annotaatio on esitetty kaaviona kuvassa 5.



Kuva 5. *Ominaisuuden kautta dokumenttiin liitetty annotaatio*

Liittämällä annotaatio dokumenttiin ominaisuuden kautta, voidaan ominaisuuden avulla ilmaista annotaation ja dokumentin välisen suhteen merkitys. Suhdetta voidaan kuvailla antamalla ominaisuudelle nimeksi esimerkiksi ”kuvailee”, ”liittyy”, ”luokittelee” jne. Tämän avulla annotaatioita on myöhemmin mahdollista luokitella mm. sen perusteella, millä tavalla ne kytkeytyvät annotoituihin dokumentteihin.

Taulukossa 2 on vertailtu edellä kuvattujen viittaustapojen eroja erilaisissa annotaatioiden tuottamiseen ja käyttöön liittyvissä tilanteissa. Jokaisen käyttötilanteen kohdalla on merkitty vihreällä pohjalla se viittaustapa, joka useimmissa tilanteissa voidaan katsoa olevan annotaatioiden hallinnan, muodostamisen tai käytön kannalta edullisempi.

Taulukko 2. *Viittausmenetelmien vertailua*

	Dokumenttien luokittelu	**Ominaisuudesta viittaaminen**
Annotoidun dokumentin URL muuttuu:	Luodaan uusi annotaatio, jonka URI on uusi URL ja tehdään vanhasta annotaatiosta viittaus siihen.	Muutetaan dokumenttiin viittaavan ominaisuuden arvoa.
Eri dokumenttien annotointi samalla annotaatiolla:	Luodaan jokaiselle dokumentille oma annotaatio.	Määritellään viittauksessa käytettävälle ominaisuudelle useampi arvo.
Eri annotaatioiden liittäminen samaan dokumenttiin:	Määritellään, että dokumentti on useamman eri luokan ilmentymä.	Viitataan eri annotaatioista dokumenttiin.
Kaikkien tiettyyn dokumenttiin liittyvien annotaatioiden haku:	Haetaan annotaatiot, joiden URI on sama kuin dokumentin.	Etsitään kaikki annotaatiot, joissa viitataan dokumenttiin.
Kaikkien tiettyyn annotaatioluokkaan liittyvien dokumenttien (URL) haku:	Haetaan kaikki annotaatioluokan ilmentymät.	Haetaan kaikki annotaatioluokan ilmentymät ja niistä dokumenttiin viittaavan ominaisuuden arvot.

Taulukosta 2 havaitaan, että ominaisuuden avulla tehdyt viittaukset annotoituun dokumenttiin ovat yleensä dokumenttien luokitteluun verrattuna joustavampia. Jos esimerkiksi annotoidun dokumentin URL muuttuu, tulee dokumentin luokittelussa luoda uusi vastaava annotaatio ja huolehtia siitä, että vanhaan annotaatioon mahdollisesti muissa resursseissa olevat viittaukset ohjataan uuteen. Ominaisuudesta viittaamisessa taas riittää, että muutetaan viittaavan ominaisuuden arvona oleva vanha URL uudeksi. Dokumentteja luokittelevat annotaatiot ovat puolestaan usein helpompia annotaatioiden haun kannalta, koska annotaation tunnisteena (URI) toimii dokumentin URL.

3.5 Annotoinnin automatisointi

Suurien tietoaineistojen annotoinnissa tarvitaan automatisoituja menetelmiä, jotta annotoinnissa saavutettaisiin riittävä tehokkuus ja taloudellisuus (Reeve & Han, 2005; Dill ym. 2003). Automaatiolla viitataan annotaatiosovellusten yhteydessä tapaan, jolla sovellus tukee ja annotoijan näkökulmasta helpottaa annotaatioiden muodostamista kuvailtavan tekstiaineiston pohjalta. Eri sovellukset tarjoavat tyypillisesti vähintään annotaatioissa käytettävän merkkauksen automaattisen muodostamisen annotoijan tekemien valintojen pohjalta, jolloin annotoijan ei tarvitse tuntea annotaatioissa käytettävän merkkauskielen syntaksia. Automaation avulla annotoitavasta tekstistä voidaan myös tunnistaa erilaisia entiteettejä sekä käsitteitä ja tuottaa annotaatioita automaattisesti niiden pohjalta. Koska täysin automaattinen annotointi on ratkaisematon ongelma, monet nykyisistä annotointijärjestelmistä keskittyvät niin sanottuun *puoliautomaattiseen annotointiin* (Reeve & Han, 2005). Siinä annotoijan työtä helpotetaan mm. erilaisilla

tiedoneristämismenetelmillä (Information Extraction), joilla tarkoitetaan jäsennellyn esityksen, kuten esimerkiksi tietokannan muodostamista tekstistä valitusta tiedosta (Grishman, 1997). Annotoinnin apuna voi olla myös *koneoppiminen* (Machine Learning), jossa annotointisovellus opettelee manuaalisesti tehtävän annotoinnin perusteella tunnistamaan annotoitavasta tekstistä esimerkiksi nimettyjä entiteettejä.

Yleisimpiä automaattisissa annotointijärjestelmissä hyödynnettäviä menetelmiä ovat erilaiset säännölliset lauseet (*regular expression*), joiden avulla voidaan etsiä säännöllisiä rakenteita noudattavia merkkijonoja annotoitavasta tekstistä. Esimerkiksi päivämäärä, joka on muotoa vvvv-kk-pp (esim. 2006-07-13), voidaan tunnistaa seuraavalla säännöllisellä lauseella:

```
(19|20)\d\d-(0[1-9]|1[012])-(0[1-9]|[12][0-9]|3[01])
```

Lauseessa ilmaistaan, että etsittävä vuosiluku alkaa joko numeroilla 19 tai 20 (`(19|20)`) ja niiden jälkeen tulee mitkä tahansa kaksi numeroa (`\d\d`) ja väliviiva (`-`). Tämän jälkeen ilmaistaan vastaavasti, että kuukausi on jokin numero väliltä 01-12 ja päivä numero väliltä 01-31.

Tekstiaineistoa voidaan annotoida automaattisesti myös niin sanotuilla ohjatuilla (*supervised*) tiedoneristämisjärjestelmillä, kuten Amilcarella (Ciravegna & Wilks, 2003). Tällaisissa järjestelmissä annotointi perustuu siihen, että ne opettelevat aluksi annotoitavien asioiden tunnistamista valmiiksi annotoiduista teksteistä. Oppimisen tavoitteena on muodostaa tiedoneristämissääntöjä, joiden tehtävä muistuttaa edellä kuvattuja säännöllisiä lauseita. Eristämissääntöjen perusteella varsinaisesta annotoitavasta tekstiaineistosta voidaan tunnistaa erilaisia asioita, kuten nimettyjä entiteettejä. Tällaisissa menetelmissä ongelmana on mm. hyvän opetusaineiston löytäminen, koska sen tulisi mahdollisimman hyvin edustaa rakenteeltaan sisällöltään koko annotoitavaa aineistoa (Uren ym. 2006).

Annotoinnin automatisoinnilla on rajansa ja niiden ylittäminen johtaa helposti joko puuttuviin tai virheellisiin annotaatioihin. Puuttuvien annotaatioiden yhteydessä voidaan puhua myös matalasta saannista (low recall) ja virheellisten kohdalla vastaavasti matalasta tarkkuudesta (low precision). Kun jompaakumpaa edellisistä parannetaan, on yleensä vaarana, että toinen huononee (Uren ym. 2006). Automaattisten menetelmien rajoituksia kuvaa esimerkiksi se, että niiden avulla on vaikeaa *identifioida* annotoitavasta tekstistä automaattisesti tunnistettuja entiteettejä, kuten ihmisiä ja erilaisia paikkoja. Esimerkiksi tekstistä tunnistettua ihmisen nimeä on vaikeaa automaattisesti yhdistää juuri siihen henkilöön, johon nimi viittaa. Yhtälailla on vaikeaa määritellä, mihin tiettyyn maantieteelliseen paikkaan tekstissä viitataan, jos saman samannimisiä paikkoja on olemassa useita. Automaattisten menetelmien toinen merkittävä ongelma on erilaisten semanttisten suhteiden tunnistaminen (Cimiano ym. 2003). Tiedoneristämismenetelmät osaavat tyypillisesti tunnistaa tekstistä erilaisia entiteettejä ja käsitteitä, mutta eivät osaa päätellä niiden välisiä suhteita. Tyypillinen esimerkki on teksti, joka sisältää useita nimiä ja puhelinnumeroita. Automaattinen tiedoneristämisjärjestelmä ei välttämättä pysty

yhdistämään numeroita oikeisiin henkilöihin (Uren ym. 2006). Edellä mainittujen ongelmien johdosta automatisoidut annotointimenetelmät vaativat yleensä myös ihmisen suorittamia toimenpiteitä, joihin voi kuulua esimerkiksi automaattisesti tunnistettujen entiteettien identifiointia sekä monimerkityksellisten käsitteiden erottelua. Mikäli halutaan monipuolisesti kuvailla erilaisissa dokumenteissa esiintyviä semanttisia suhteita, on annotointia yleensä suoritettava manuaalisilla menetelmillä.

4 ANNOTAATIOSOVELLUKSET

Tässä luvussa esitellään eräitä semanttisen webin annotaatiosovelluksia sekä työkaluja, joita voidaan käyttää annotoinnin yhteydessä. Mikäli mahdollista, esiteltäviä sovelluksia on testattu käytännössä. Luvun lopussa esitetään yleisiä annotaatiosovellusten toimintaan ja ominaisuuksiin liittyviä vaatimuksia.

4.1 Katsaus ontologiaperustaisiin annotaatiosovelluksiin

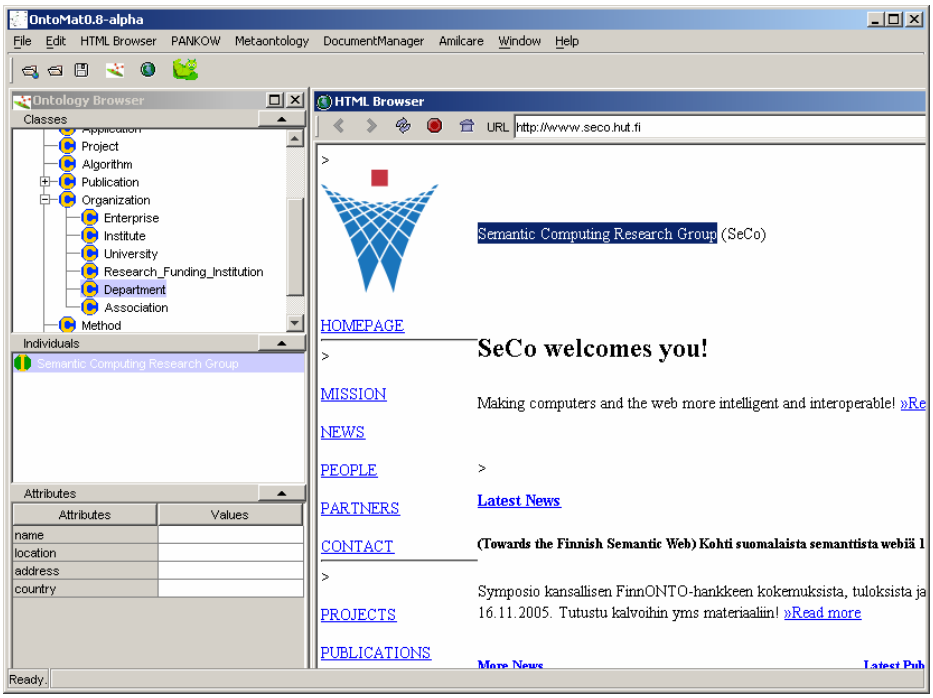
4.1.1 Yleistä

Ontologiapohjaiset annotaatiosovellukset on pääasiassa suunniteltu ontologisten merkkausten tuottamiseen ja ylläpitoon www-sivuille (Corcho, 2006). Niiden tarkoituksena on toisin sanoen tarjota työkaluja annotaatioiden muodostamiseen, tallentamiseen ja ylläpitoon. Annotaatiosovellukset eroavat toisistaan mm. sen mukaan, minkälaista informaatiota niillä kuvataan, minkälaista merkkauskieltä ne tuottavat ja minkälaisen automaation tason ne tarjoavat (Kettler ym. 2005).

4.1.2 Ont-O-Mat -annotaatioeditori

Ont-O-Mat[15] on S-CREAM-annotointimallin toteuttava vapaasti saatavilla oleva annotaatiosovellus (Handschuh ym. 2002). Sillä voidaan annotoida ontologioihin pohjautuen www-sivuja ja hyödyntää annotoinnissa myös automaattista tiedoneristämistä. Ont-O-Mat:iin voidaan ladata käyttäjän valitsemia OWL-muotoisia ontologioita ja myös niiden yksinkertainen muokkaaminen on mahdollista. Kuvassa 6 on esitetty Ont-O-Mat:in käyttöliittymä. Käyttöliittymän vasemmassa laidassa näkyy ylhäällä sovellukseen ladatun ontologian luokkahierarkia. Sen alla on näkymä valitusta luokasta luotuihin ilmentymiin ja edelleen ilmentymänäkymän alla luokkaan liittyvät ominaisuudet. Käyttöliittymän oikealla puolella on HTML-selain, johon voidaan ladata www-sivu, jota halutaan annotoida.

[15] http://annotation.semanticweb.org/ontomat/



Kuva 6. *Ont-O-Mat annotaatioeditori*

Ont-O-Mat tarjoaa annotoitavan dokumentin näkökulmasta kolme erilaista annotointitapaa. Ensimmäisessä annotointitavassa luodaan käytössä olevan ontologian pohjalta annotoitavaa resurssia kuvailevia subjekti-predikaatti-objekti-lauseita. Lauseet muodostetaan luomalla ontologian luokista ilmentymiä ja määrittelemällä arvot ilmentymiin liittyville ominaisuuksille. Arvoina voivat toimia joko literaalit tai toiset ontologiasta muodostetut ilmentymät. Toisessa annotointitavassa annotaatioita muodostetaan kuvailtavassa resurssissa (www-sivu) olevaa tekstiä hyödyntäen. Annotoija voi muodostaa ontologian luokasta ilmentymän maalaamalla hiirellä dokumentista tekstikohdan ja vetämällä sen haluamansa luokan päälle, jolloin ilmentymä muodostetaan. Luontihetkellä ilmentymään liitetään lisäksi XPointer:lla[16] määritelty viite valittuun tekstikohtaan sekä käyttäjän valinnan mukaan asetetaan maalattu teksti automaattisesti jonkin ilmentymälle kuuluvan ominaisuuden literaaliarvoksi. Kolmannessa annotointitavassa annotaatioiden muodostaminen on kytketty www-sivujen luontiin. Ont-O-Mat sisältää HTML-editorin, jonka avulla käyttäjä voi luoda uuden web-sivun ja samalla liittää siihen ontologian pohjalta muodostettuja annotaatioita.

Manuaalisen annotoinnin lisäksi Ont-O-Mat tukee myös www-sivujen puoliautomaattista annotointia. Automaattisuus perustuu Amilcare-tiedoneristämisjärjestelmään, jonka avulla annotoitavista sivuista voidaan tunnistaa mm. sanaluokkia, lauseita sekä nimettyjä entiteettejä.

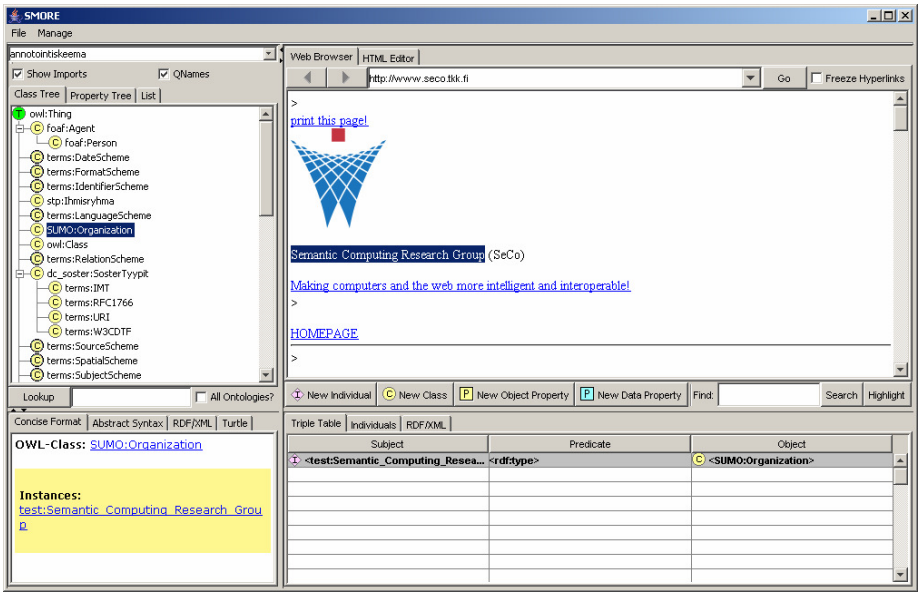
[16] XPointer on järjestelmä, jonka avulla voidaan osoittaa haluttuihin elementteihin XML-pohjaisessa dokumentissa (ks. http://www.w3.org/XML/Linking)

4.1.3 MnM-annotaatioeditori

MnM (Vargas-Vera ym. 2002) on vapaasti saatavilla oleva annotaatioeditori, joka tukee www-sivujen automaattista ja puoliautomaattista annotointia. Sovellus yhdistää www-selaimen ontologiaeditoriin ja tarjoaa avoimia sovellusrajapintoja, joiden avulla se voidaan kytkeä ontologiapalvelimiin sekä tiedoneristämistyökaluihin. Automatisoidussa annotoinnissa MnM käyttää Amilcare-järjestelmää, joka tulee MnM:n perusversion mukana. MnM muistuttaa toiminnoiltaan ja käyttöliittymältään pitkälti Ont-O-Mat:ia. Ont-O-Mat:ssa annotaatiot liitetään annotoitavien dokumenttien yhteyteen, kun taas MnM:ssä ne tallennetaan dokumenttien lisäksi erilliselle palvelimelle.

4.1.4 SMORE-annotaatioeditori

SMORE (Kalyanpur ym. 2005) on vapaasti saatavilla oleva annotaatioeditori, jossa pyritään yhdistämään www-sivujen luonti ja annotointi. Se sisältää graafisen HTML-editorin, jonka avulla voidaan luoda tavallisia HTML-sivuja. Ont-O-Mat:in tapaan SMORE:n käyttöliittymä (kuva 7) koostuu vasemmassa reunassa olevasta ontologianäkymästä ja sen oikealla puolella olevasta ikkunasta, jossa näkyy annotoitava www-sivu. Mikäli halutaan luoda uusi www-sivu, voidaan annotoitavan sivun näyttö vaihtaa HTML-editoriksi. Annotoitavan sivun alapuolella näkyy taulukko, jossa esitetään subjekti-predikaatti-objekti-muodossa sivuun liitetyt annotaatiot.



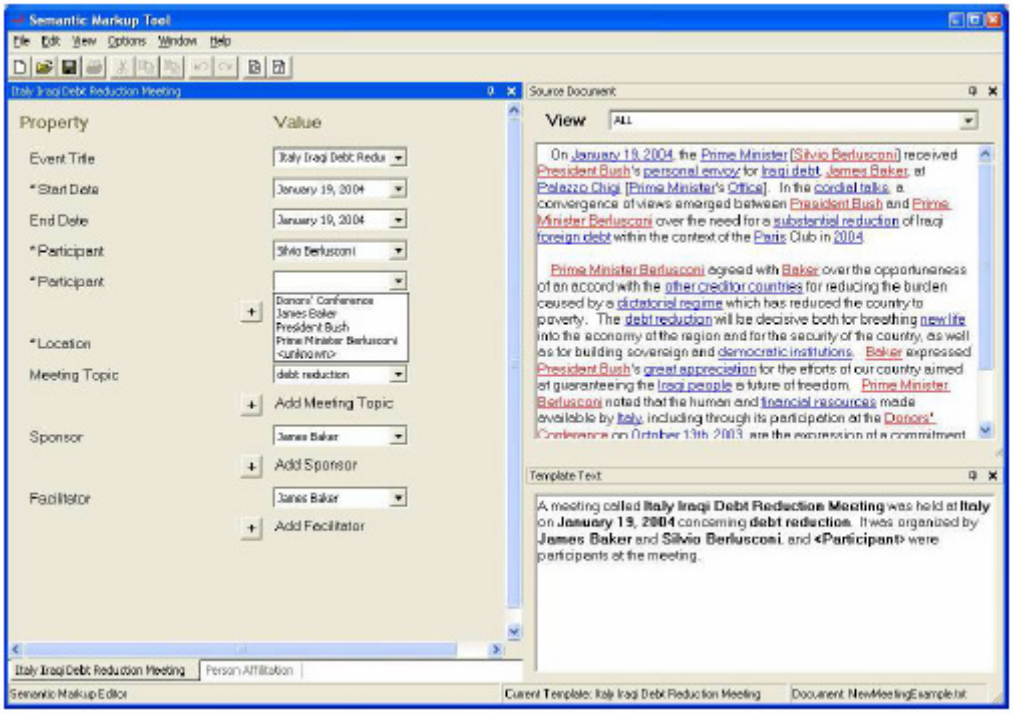
Kuva 7. *SMORE annotaatioeditori*

Annotaatioita voidaan muodostaa maalaamalla annotoitavasta www-sivusta tekstikohtia ja vetämällä niitä ontologiassa olevien luokkien päälle, jolloin kyseisestä luokasta luodaan ilmentymä. Kuvassa 7 on maalattu annotoitavasta www-sivusta (http://www.seco.tkk.fi) teksti ”Semantic Computing Research Group” ja määritelty sen olevan instanssi luokasta ”SUMO:Organization”. Muodostettu annotaatio näkyy kuvassa annotaatiotaulukossa, jossa subjektina on maalattu teksti, predikaattina `<rdf:type>` ja objektina valittu luokka. Annotaatiota voidaan muodostaa myös vetämällä www-sivusta maalattuja tekstejä suoraan

annotaatiotaulukon johonkin sarakkeeseen. Tällöin maalatut tekstikohdat tulee vielä erikseen kiinnittää johonkin ontologiassa määritettyyn käsitteeseen, jotta annotaatio olisi formaalisti määritelty.

4.1.5 Semantic Markup Tool -annotaatioeditori

Semantic Markup Tool (SMT) on kaupallinen sovellus, joka tukee www-sivujen puoliautomaattista annotointia (Kettler ym. 2005). SMT:ssa annotointi pohjautuu XML:lla kuvattuihin annotaatioskeemoihin, joissa määritellään erilaiset annotaatiotyypit ja niihin liittyvät ominaisuudet. SMT osaa tunnistaa automaattisesti tekstistä erilaisia entiteettejä ja ehdottaa niitä skeemassa määriteltyjen ominaisuuksien arvoiksi ominaisuuksien arvoalueiden perusteella. SMT:n käyttöliittymä on esitetty kuvassa 8. Vasemmassa reunassa näkyy skeeman perusteella muodostettu lomake, jonka kentät perustuvat skeemasta valitun annotaatioluokan ominaisuuksiin. Lomakkeen oikealla puolella näkyy annotoitavasta www-sivusta poimittu teksti, johon on sinisellä ja punaisella korostettu automaattisen tiedoneristäjän tunnistamat entiteetit. Tekstin alapuolella näkyy annotaation pohjalta muodostettava tiivistelmä, joka kuvaa annotoitavaa dokumenttia.



Kuva 8. *Semantic Markup Tool annotaatioeditori (Kettler ym. 2005)*

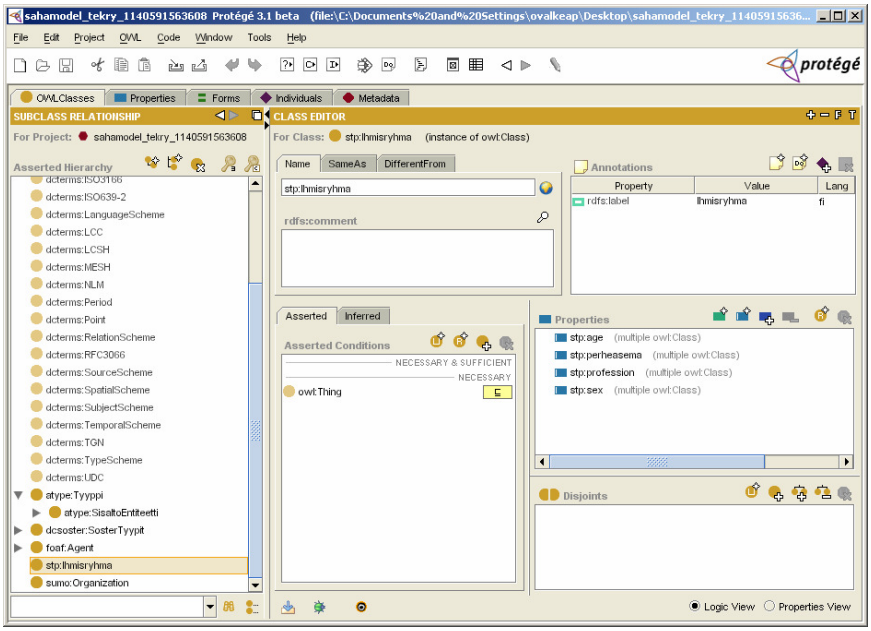
SMT:n annotointiprosessi etenee siten, että ensin annotoitava dokumentti muutetaan alkuperäisestä formaatistaan (esim. HTML) puhtaaksi tekstiksi ja indeksoidaan avainsanojensa perusteella. Tämän jälkeen teksti prosessoidaan erilaisilla tiedoneristäjillä, jotka etsivät siitä nimettyjä entiteettejä sekä määrämuotoisia merkkijonoja, kuten päivämääriä. Tekstistä tunnistetut entiteetit lisätään automaattisesti sovelluksen tietämyskantaan ja niille pyritään määrittämään luokkavastaavuuksia ontologioista. Seuraavaksi tunnistettuja luokkia verrataan valitussa skeemassa olevien ominaisuuksien

arvoalueina oleviin luokkiin ja tämän perusteella ehdotetaan entiteettejä sopivien ominaisuuksien arvoiksi. Annotoijan tehtävänä on hyväksyä järjestelmän tekemät ehdotukset tai muuttaa niitä, jos ne ovat vääriä. Annotoija voi määritellä annotaation ominaisuuksia lomakkeeseen tai skeeman pohjalta muodostettuun dokumenttia kuvaavaan tiivistelmään, jossa on puuttuvia sanoja, jotka annotoija korvaa dokumentista tunnistetuilla entiteeteillä. Annotaatio tallennetaan OWL-muodossa irrallaan varsinaisesta dokumentista.

4.2 Muita sovelluksia

4.2.1 Protégé-ontologiaeditori

Protégé[17] on vapaasti saatavilla oleva graafinen työkalu ontologioiden editointiin (Noy ym. 2000). Se tarjoaa monipuoliset työkalut RDF-pohjaisen tiedon hallintaan ja tukee erilaisia laajennuksia (plugin), joiden kautta voidaan lisätä tuki mm. OWL-semantiikalle. Protégé on ensisijaisesti suunniteltu ontologioiden kehittämistä varten, mutta sillä voidaan suorittaa myös erilaisten resurssien annotointia. Protégén perusversion käyttöliittymä ei tue annotointia samalla tavalla kuin varsinaiset annotaatiosovellukset. Se sisältää paljon varsinaisen annotoinnin kannalta vähemmän tarpeellisia ontologioiden muokkaamiseen liittyviä toimintoja, eikä se tarjoa esimerkiksi mahdollisuutta katsella annotoitavaa dokumenttia. Protégéta voidaan kuitenkin käyttää annotoinnissa sekä apuvälineenä mm. annotaatioskeemojen luonnissa. Kuvassa 9 on esitetty Protégén käyttöliittymä, johon on ladattu OWL-muotoinen ontologia.



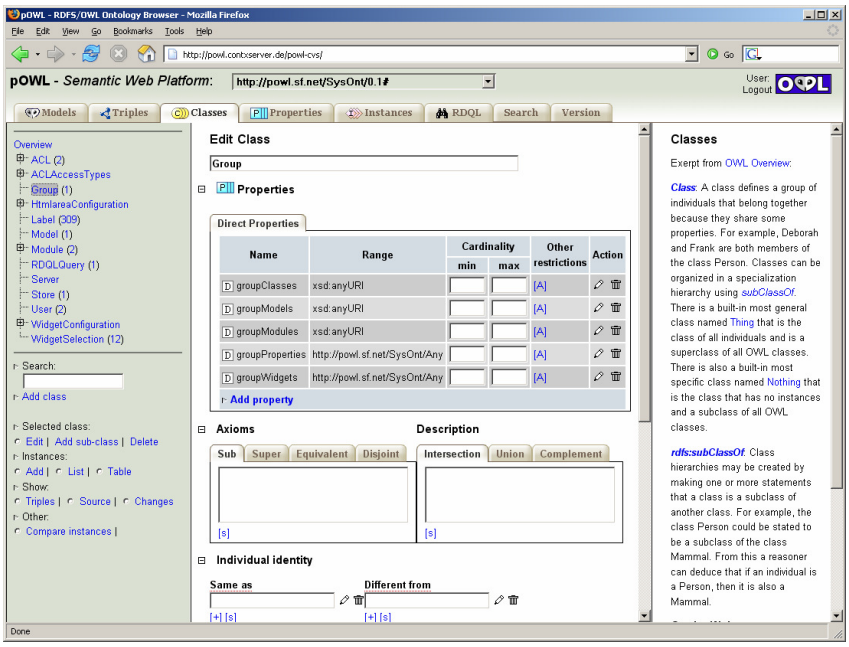
Kuva 9. *Protégé-ontologiaeditori*

[17] http://protege.stanford.edu/

4.2.2 Ontologioiden hallintajärjestelmä pOWL

pOWL[18] (Auer, 2005) on PHP-pohjainen[19], vapaaseen lähdekoodiin perustuva ontologioiden hallintajärjestelmä. Se tukee RDFS- ja OWL-tyyppisten tietämyskantojen tallentamista, muokkaamista, kyselyä, versiointia ja sarjallistamista web-ympäristössä. pOWL:in arkkitehtuuri rakentuu sovellusrajapinnoista, joilla voidaan käsitellä RDF-, RDFS- ja OWL-muotoista tietoa, sekä rajapinnasta, jolla voidaan tehdä web-sovelluksia näiden päälle. pOWL:ssa käsiteltävä tieto tallennetaan SQL-pohjaiseen (Structured Query Language) relaatiotietokantaan ja se sisältää valmiin käyttöliittymän tietokantaan tallennettujen tietomallien muokkaamiseen.

pOWL:in selainpohjainen käyttöliittymä (kuva 10) on järjestetty välilehdille, jotka tarjoavat erilaisia näkymiä alla olevaan tietomalliin. Välilehdillä voidaan tarkastella mm. järjestelmään ladattujen ontologioiden luokkahierarkioita, ominaisuuksia ja ilmentymiä sekä muokata niitä halutulla tavalla. Käyttöliittymässä voidaan tehdä hakuja tietomalleihin ja tarkastella niiden versiohistoriaa, johon tallennetaan kaikki ontologioihin tehdyt muutokset.



Kuva 10. *Ontologioiden hallintajärjestelmä pOWL*

pOWL muistuttaa monilta toiminnoiltaan Protégé-editoria. Sitä ei Protégén lailla ole ensisijaisesti suunniteltu annotointia varten, mutta sen tarjoamien rajapintojen pohjalta on mahdollista toteuttaa web-pohjaisia annotointisovelluksia. pOWL:in RDF-pohjaisen tiedon käsittelyyn suunnitellut sovellusrajapinnat muistuttavat monilta osin Javan Jenaa[20] ja tarjoavat näin monia mahdollisuuksia semanttiseen webiin liittyvien sovellusten

[18] http://powl.sourceforge.net/

[19] http://www.php.net/

[20] http://jena.sourceforge.net/

laatimiseen. pOWL:in etuna on sen hyödyntämien tekniikoiden (PHP ja MySQL[21]) laaja levinneisyys.

4.2.3 ONKI-ontologiapalvelin

ONKI (Komulainen ym. 2005) on Tekesin FinnONTO-hankkeessa kehitetty kansallisten ontologioiden kehitysympäristö. Sen tavoitteena on tukea eri tahojen yhteistyössä suorittamaa ontologiakehitystä ja toisistaan riippuvien ontologioiden uudelleenkäyttöä. ONKI:n ominaisuuksiin kuuluu ontologioiden muutoksenhallinta ja versiointi sekä selainkomponentti, joka tarjoaa ontologioiden haku- ja käyttöpalveluja Web Services-arkkitehtuurin mukaisesti.

ONKI ei ole varsinainen annotaatiojärjestelmä, mutta se toimii tärkeänä apuvälineenä annotoijille tarjoamalla palveluita, joiden avulla voidaan helposti liittää annotaatioihin referenssiontologioissa määriteltyjä käsitteitä. ONKI toimii siis eräänlaisena viitekirjastona, josta annotoija voi hakea haluamiaan käsitteitä ja liittää niitä osaksi tekemiään annotaatioita. Annotoijien on myös mahdollista lähettää ONKI:in luomiaan ilmentymiä, jolloin ne ovat muiden ONKI:a hyödyntävien annotoijien käytettävissä.

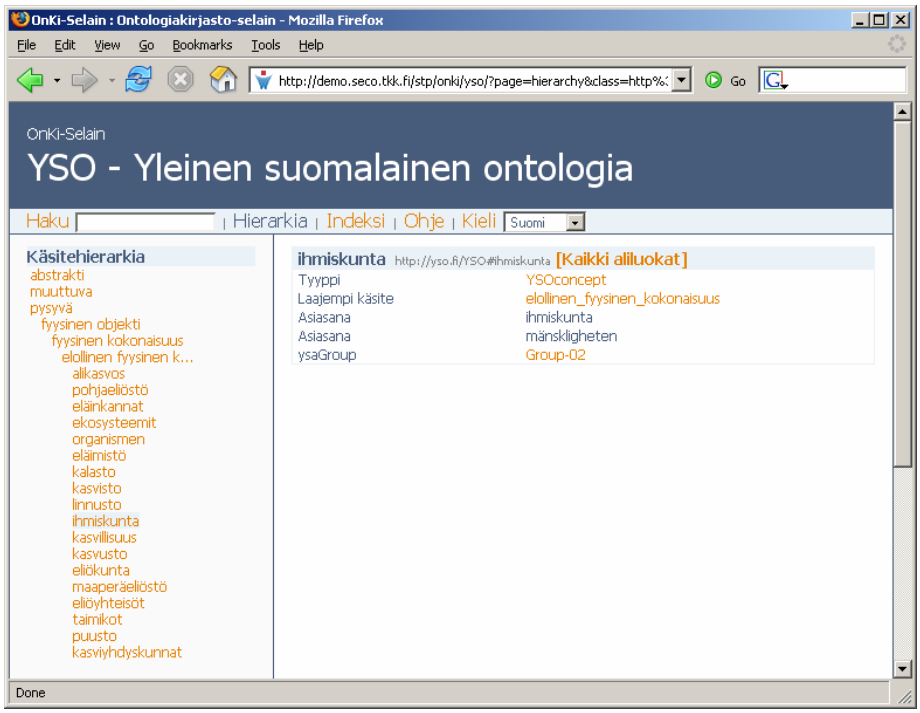
Kuvassa 11 on esitetty ONKI:n www-selaimella käytettävä ontologioiden selailu- ja hakukäyttöliittymä, johon on ladattu yleinen suomalainen ontologia (YSO). Ulkoiset sovellukset voivat kytkeytyä ONKI:in SOAP-protokollaa[22] käyttämällä ja selainpohjaiset web-sovellukset myös Javascript- sekä Ajax[23]-tekniikoilla. ONKI:n tarjoaman keskitetyn ontologiapalvelun ansiosta annotaatiosovelluksille on aina tarjolla ajan tasalla olevat versiot ontologioista.

---

[21] http://www.mysql.com/

[22] SOAP (Simple Object Access Protocol) on protokolla, jonka avulla voidaan välittää XML-pohjaisia viestejä tietokoneverkoissa, yleensä HTTP:tä käyttäen.

[23] Ajax (Asynchronous JavaScript And XML) on tekniikka, jolla voidaan toteuttaa interaktiivisia web-sovelluksia.



Kuva 11. *ONKI-selain*

4.3 Annotaatiosovellusten arviointia

Taulukossa 3 on yhteenveto edellä esitettyjen annotointisovellusten perusominaisuuksista. Taulukossa on vertailun vuoksi mukana myös tämän diplomityön pohjalta toteutettu Saha-annotaatioeditori.

Taulukko 3. *Annotaatiosovellusten ominaisuuksien vertailua*

Sovellus	**OntoMat**	**MnM**	**SMORE**	**SMT**	**Saha**
Versio	0.8	2.1	5.0	1.0	2.3
Alusta	Java	Java	Java	MS .NET (C#)	Java (palvelimella)
Käyttöliittymä	Sovellus	Sovellus	Sovellus	Sovellus	www-selain
Annotoitavien dokumenttien formaatti	HTML	HTML, teksti	HTML, teksti, sähköposti, kuva	HTML	Mikä tahansa dokumentti, johon voidaan viitata URI:lla
Tuotettujen annotaatioiden formaatti	OWL	RDFS, DAML+OIL	OWL	OWL	OWL
Ontologioiden muokkaus	kyllä	ei	kyllä	ei	kyllä (uusien aliluokkien ja ilmentymien luonti)
Tuki annotointi-skeemoille	ei	ei	ei	kyllä	kyllä
Pääsy ulkoiseen tietämyskantaan	ei	kyllä	ei	kyllä	kyllä (ONKI-palvelin)
Annotaatiot tallennetaan	Annotaatio-palvelimelle, tai osaksi annotoituja HTML-sivuja	Osaksi annotoituja HTML-sivuja	Tekstitiedostoon	-	Palvelimelle tietokantaan

Esitellyt sovellukset ovat monien semanttisen webin sovellusten tapaan kokeellisia, eikä niillä tuotetuille annotaatioille ole olemassa vielä tällä hetkellä kovin konkreettisia käyttökohteita. Näistä seikoista johtuen sovellusten ominaisuudet ja käytettävyys eivät yllä vielä tasolle, jota esimerkiksi kaupallisilta sovelluksilta tyypillisesti vaaditaan. Sovellusten yhtenä merkittävimmistä heikkouksista voidaan pitää niiden huonoa soveltuvuutta sellaisille käyttäjille, jotka eivät tunne tarkemmin semanttisen webin tekniikoita ja erikoiskäsitteitä. Jotta semanttista webiä varten onnistuttaisiin tuottamaan mahdollisimman paljon laadukasta metatietoa, tulisi mahdollisimman suurella joukolla webin käyttäjiä olla käytössään yksinkertaisia työvälineitä semanttisesti rikkaiden annotaatioiden muodostamiseen. Edellä esiteltyjen sovellusten heikkoutena voidaan pitää lisäksi sitä, että suurin osa niistä tulee asentaa paikallisesti, vaikka loppukäyttäjän näkökulmasta olisi usein helpompaa, jos ne toimisivat keskitettyinä web-sovelluksina.

Esitellyistä sovelluksista SMT tarjoaa yksinkertaisimman käyttöliittymän, jossa annotaatioiden määrittely tapahtuu annotaatioskeeman perusteella muodostettavalla lomakkeella. Skeemassa olevien ominaisuuksien arvot on helppo määrittää lomakkeen kenttiin, mikäli annotoitavasta tekstistä automaattisesti tunnistetut entiteetit täsmäävät niihin. SMT:ssä voidaan kuitenkin käyttää vain hyvin yksinkertaisia skeemoja, mikä on selkeä rajoite ohjelman käytettävyydelle monimutkaisempien semanttisten rakenteiden kuvailussa. Koska sovellusta ei pystytty testaamaan, ei myöskään voitu selvittää miten hyvin sen suorittama automaattinen tiedoneristäminen toimii.

Ont-O-Mat:in, MnM:n ja SMORE:n käyttö saattaa olla hyvin vaikeaa sellaisille annotoijille, jotka eivät tunne semanttisen webin tekniikoita, kuten esimerkiksi RDF-kieltä. Ont-O-Matin avulla pystytään muodostamaan tarvittaessa monipuolisia semanttisia kuvauksia, mutta hyvän ilmaisuvoiman hintana on monimutkainen käyttöliittymä. Dokumenttien kuvailu muistuttaa näissä sovelluksissa vapaata annotointia, koska annotaatioita ei muodosteta skeeman perusteella luodun lomakkeen pohjalta kuten SMT:ssä. Sovellusten käyttöliittymissä erilaisiin ontologisiin resursseihin viitataan URI-tunnisteilla sen sijaan, että käytettäisiin ihmisille tarkoitettujen otsikoita. SMORE:ssa käytössä oleva tapa muodostaa annotaatiot RDF-kielessä käytetyn subjekti-predikaatti-objekti-mallin avulla tarjoaa joustavan tavan annotaatioiden muodostamiseen, mutta edellyttää hyvää perustuntemusta RDF:sta.

Ont-O-Mat:ssa ja SMORE:ssa on pyritty sisäänrakennetulla web-editorilla yhdistämään web-sivujen luonti ja annotointi yhdeksi tapahtumaksi. Editorin avulla käyttäjä voi luoda HTML-sivun ja tuottaa samalla sivua kuvailevat annotaatiot. Dokumenttien tuottamisen ja annotoinnin yhdistäminen on ajatuksena hyvä, mutta käytännössä web-sivuja laaditaan enää hyvin harvoin sellaisilla staattisten sivujen luontiin tarkoitetuilla editoreilla, joita Ont-O-Mat ja SMORE tarjoavat käyttäjilleen. Ne ovat käyttökelpoisia lähinnä yksittäisten, hyvin yksinkertaisten web-sivujen luonnissa ja annotoinnissa.

4.4 Vaatimuksia annotaatiosovelluksille

Semanttisen webin sisältöjen kuvailutyön pitää tulevaisuudessa olla mahdollista myös niille webin käyttäjille, joilla ei ole teknistä tuntemusta webistä. Tästä näkökulmasta katsottuna tarjolla pitäisi olla annotointisovelluksia, joita on yksinkertaista käyttää ja joiden yhteydessä annotoija mahdollisimman vähän kohtaa semanttiseen webiin liittyviä teknisiä erikoiskäsitteitä. Annotointisovellusten kehittämisessä tulisi toisin sanoen ottaa huomioon myös ”tavalliset” webin käyttäjät annotaatioiden tuottajina (Vargas-Vera ym. 2002). Tässä luvussa esiteltyjä annotaatiosovelluksia tutkittaessa nousi esiin seuraavia ominaisuuksia, joita sovellusten tulisi tarjota, jotta niillä voitaisiin tuottaa mahdollisimman yksinkertaisella tavalla semantiikaltaan riittävän monipuolisia annotaatioita:

- **Merkkauskieliin ja ontologioihin liittyvien erikoiskäsitteiden piilottaminen**. Mikäli näiden esittäminen ei ole annotoinnin kannalta välttämätöntä, niitä ei tulisi näyttää sovelluksessa. Sovelluksessa olisi hyvä välttää myös monimutkaisten luokkahierarkioiden esittämistä ja tarjota annotoijalle sen sijaan mahdollisuus etsiä niissä olevaa tietoa monipuolisesti toteutettujen hakujen avulla.

- **Skeemojen hyödyntäminen annotoinnissa**. Annotaatioskeemoilla voidaan kuvata annotaation rakenne ja määritellä sitä kautta asiat, joita annotoitavasta dokumentista halutaan kuvailla. Tämä helpottaa annotoijan työtä ja parantaa muodostettavien annotaatioiden yhdenmukaisuutta ja yhteentoimivuutta. Skeemaa hyödyntämällä annotointisovellus voidaan myös suunnitella geneeriseksi, koska skeeman avulla voidaan ohjata sovelluksen käyttöliittymän muodostamista.

- **Annotoinnin jakaminen**. Annotaatiosovelluksen tulisi tukea annotaatioiden hajautettua tuottamista, jossa eri annotoijat voivat viitata yhteisesti määriteltyihin resursseihin ja parantaa näin annotaatioiden semanttista yhteensopivuutta.

- **Web-pohjaisuus**. Jotta annotoinnin jakaminen usean annotoijan kesken olisi helppoa ja kynnys annotaatiosovelluksen käyttämiseen matala, on annotointisovellus käytännöllistä toteuttaa web-sovelluksena, joka asettaa mahdollisimman vähän vaatimuksia käyttäjän laitteistolle.

Edellä esitettyjen ominaisuuksien lisäksi annotaatiosovelluksen suunnittelussa tulisi huomioida myös seuraavia, Urenin ym. (2006) määrittelemiä yleisiä perusvaatimuksia annotointijärjestelmille:

- **Standardien mukaiset formaatit**. Annotoinnissa tulisi käyttää yhteisesti sovittuja standardeja, koska niiden avulla heterogeenisia aineistoja on helpompi yhdistää ja jakaa annotaatioita erilaisten käyttäjäyhteisöjen välillä. Standardeja tarvitaan sekä ontologioiden kuvailussa (esim. OWL), että annotaatioissa (esim. W3C:n annotaatioskeema)

- **Käyttäjäkeskeinen/yhteistyötä tukeva suunnittelu**. Koska harvoissa organisaatioissa on ammattimaisia annotoijia, tulisi annotaatioiden tuottamista tukea

ohjelmilla, joissa on annotointiprosessia helpottavia yksinkertaisia käyttöliittymiä ja joiden avulla annotointi muodostuu osaksi tiedontuottajien muita tehtäviä. Järjestelmien tulisi myös tukea käyttäjien välistä yhteistyötä, koska tietoa tuottaa tyypillisesti joukko eri alojen asiantuntijoita.

- **Ontologioiden käyttö**. Sopivien ontologiaformaattien lisäksi annotaatiojärjestelmien tulisi tukea useampien eri ontologioiden käyttöä sekä ontologioiden muuttumista ajan kuluessa.

- **Tuki eri dokumenttiformaateille**. Annotointijärjestelmien tulisi tukea muillakin, kuin webin alkuperäisillä merkkauskielillä (HTML ja XML) määriteltyjen dokumenttien annotointia.

- **Mukautumien dokumenttien muutoksiin**. Annotaatiojärjestelmien tulisi tukea annotaatioiden ylläpitoa dokumenttien muuttuessa. Mikäli annotaatioita kiinnitetään tiettyihin kohtiin dokumenteissa, voi annotaatioiden sisällön ylläpidon lisäksi olla tarvetta ylläpitää annotaatioiden ja dokumenttien välisiä kytköksiä.

- **Annotaatioiden tallennus**. Semanttisen webin yhteydessä annotaatiot pyritään erottamaan dokumenteista, koska annotoijilla ei välttämättä ole kytköstä dokumenttien tuottajiin. Joissain tapauksissa on kuitenkin annotaatioiden ylläpidon kannalta tarkoituksenmukaisempaa liittää ne dokumentteihin.

- **Automatisointi**. Suurien aineistomäärien kuvailussa tarvitaan automatisoituja menetelmiä. Tämän vuoksi on tärkeää, että erilaisia tiedoneristämismenetelmiä integroidaan annotaatiojärjestelmiin.

5 SAHA-ANNOTAATIOJÄRJESTELMÄ

Tässä luvussa esitellään työn puitteissa kehitetty annotaatiojärjestelmä *Saha*. Luvussa käydään ensin läpi järjestelmän tekninen toteutus ja sen jälkeen esitellään sen ominaisuudet sekä käyttöliittymä. Luvun lopussa esitetään järjestelmällä suoritettavan annotointiprosessin kokonaiskuvaus.

5.1 Suunnittelun lähtökohdat

Luvussa 3 esiteltyä skeemaperustaista annotointia sovellettiin käytännössä kehittämällä selainpohjainen annotaatiojärjestelmä *Saha* (Valkeapää & Hyvönen, 2006). Sahaa ryhdyttiin kehittämään, koska luvussa 4.1 esiteltyjen annotointisovellusten havaittiin olevan pääsääntöisesti liian monimutkaisia sellaisille käyttäjille, jotka eivät tunne semanttisen webin tekniikoita ja annotointiin liittyviä erikoiskäsitteitä. Lisäksi haluttiin kehittää sovellus, jonka avulla annotointi voitaisiin helposti jakaa useiden käyttäjien kesken ja joka toimisi käyttäjän tietokoneen sijasta web-palvelimella.

Saha on web-sovellus, jonka avulla voidaan tuottaa annotaatioskeemoihin perustuvia ontologiapohjaisia annotaatioita, joiden kohteena on tyypillisesti jokin www-sivu tai muu webissä oleva dokumentti. Sovellus on suunniteltu erityisesti erilaisten web-portaalien metadatan tuottamiseen. Sahan kehitystyön tärkeimpiin tavoitteisiin kuului tutkia, miten ontologiapohjaista annotointia voitaisiin suorittaa sellaisella abstraktiotasolla, joka mahdollisuuksien mukaan piilottaisi monimutkaiset ontologiat järjestelmän käyttäjiltä. Koska tavoitteena oli ensisijaisesti tutkia erilaisia annotointiin liittyviä menetelmiä, asetettiin ohjelman tekniselle toteutukselle ja käyttöliittymän tarkemmalle suunnittelulle pienempi painoarvo. Sahaan pyrittiin toteuttamaan luvussa 4.4 esitellyt, annotaatiosovelluksille tarpeelliseksi katsotut ominaisuudet, jotka nousivat esille erilaisia annotaatiosovelluksia tutkittaessa. Niiden pohjalta suunnittelussa otettiin tavoitteeksi seuraavat asiat:

- **Helppokäyttöisyys**. Annotoinnin tulee olla helppoa myös sellaisille annotoijille, jotka eivät tunne semanttisen webin tekniikoita. Tämän saavuttamiseksi sovelluksen tulisi, mikäli mahdollista, piilottaa käyttäjältään semanttiseen webiin, annotointiin ja ontologioihin liittyvät erikoiskäsitteet.

- **Annotoinnin hajauttaminen**. Annotaatiot tallennetaan keskitetysti palvelimelle, josta eri annotoijien on helppo hakea niitä katseltavaksi sekä muokattavaksi ja jonka kautta ne ovat hyödynnettävissä erilaisissa sovelluksissa.

- **Annotaatioskeemojen käyttö**. Annotoinnin helpottamiseksi ja semanttisesti monipuolisten annotaatioden muodostamiseksi annotaatiot luodaan skeemojen pohjalta. Skeemojen avulla voidaan myös ohjata ohjelman käyttöliittymän muodostamista.

- **Referenssiontologioiden käyttö**. Järjestelmä suunnitellaan tukemaan ONKI-palvelun käyttöä.

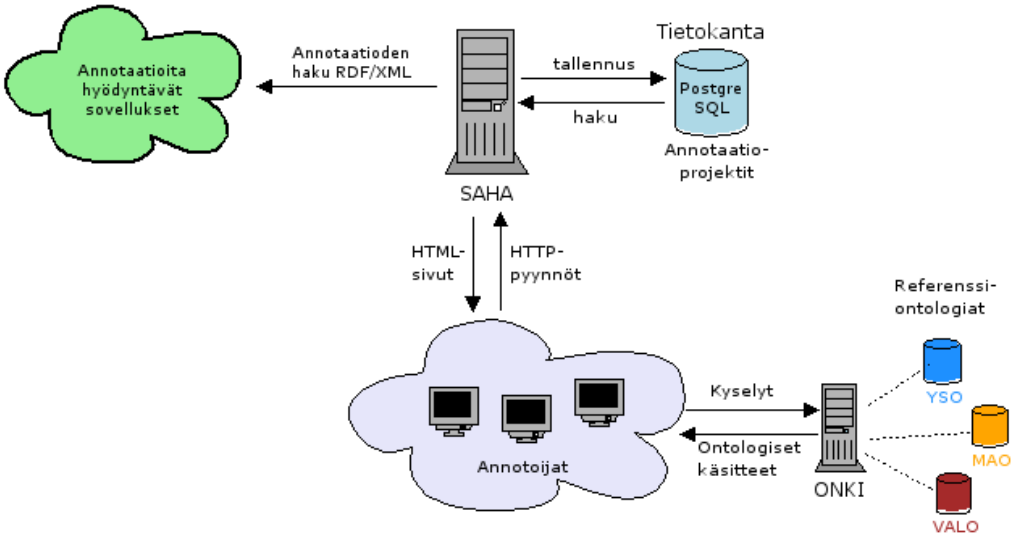
- **Selainpohjainen käyttöliittymä**. Järjestelmä toteutetaan web-sovelluksena, jolloin sen käyttöönotto on annotoijan näkökulmasta helppoa ja mahdollisimman laitteistoriippumatonta.

Sahan suunnittelussa ja toteutuksessa ei ensisijaisesti pyritty annotoinnin automatisointiin. Tavoitteena oli sen sijaan toteuttaa järjestelmä, jolla pystytään tuottamaan sellaisia semanttisesti monipuolisia annotaatioita, joiden muodostaminen ei ole mahdollista nykyisillä automaattisilla menetelmillä. Automaatiota tukevien menetelmien, kuten Poka-järjestelmän[24] hyödyntäminen jätettiin kuitenkin avoimeksi Sahan jatkokehityksessä.

5.2 Yleiskuvaus järjestelmästä

Saha on palvelimella suoritettava web-sovellus, joka muodostaa käytössä olevaan annotaatioskeeman pohjautuen annotoijan www-selaimelle lähetettävät, sovelluksen käyttöliittymänä toimivat HTML-sivut sekä huolehtii muodostettavien annotaatioiden tallentamisesta tietokantaan. Sahaan voidaan ladata useita eri annotaatioskeemoja, joista kukin muodostaa oman *annotaatioprojektinsa*. Yhdellä projektilla voi olla useampia käyttäjiä ja näin tietyn aihepiirin, organisaation tai käyttäjäryhmän annotointia voidaan jakaa useammalle annotoijalle, jotka kaikki näkevät ja pääsevät muokkaamaan projektissa tehtyjä annotaatioita. Eri annotaatioprojekteissa voidaan käyttää yhteisiä referenssiontologioiden käsitteitä kytkemällä Saha ONKI-palvelimeen. ONKI-palvelimen kautta voidaan selata referenssiontologioiden luokkahierarkioita ja hakea niissä olevia käsitteitä muodostettavien annotaatioiden ominaisuuksien arvoiksi. ONKI:n avulla eri annotaatioprojektien välillä voidaan myös haluttaessa jakaa annotaatioskeemojen pohjalta muodostettuja ilmentymiä. Sahan toimintamalli on esitetty pääpiirteittäin kuvassa 12.

[24] Poka on puoliautomaattinen annotointiympäristö, jota kehitetään FinnONTO-projektissa: http://www.seco.tkk.fi/applications/poka/



Kuva 12. *Sahan toimintamalli*

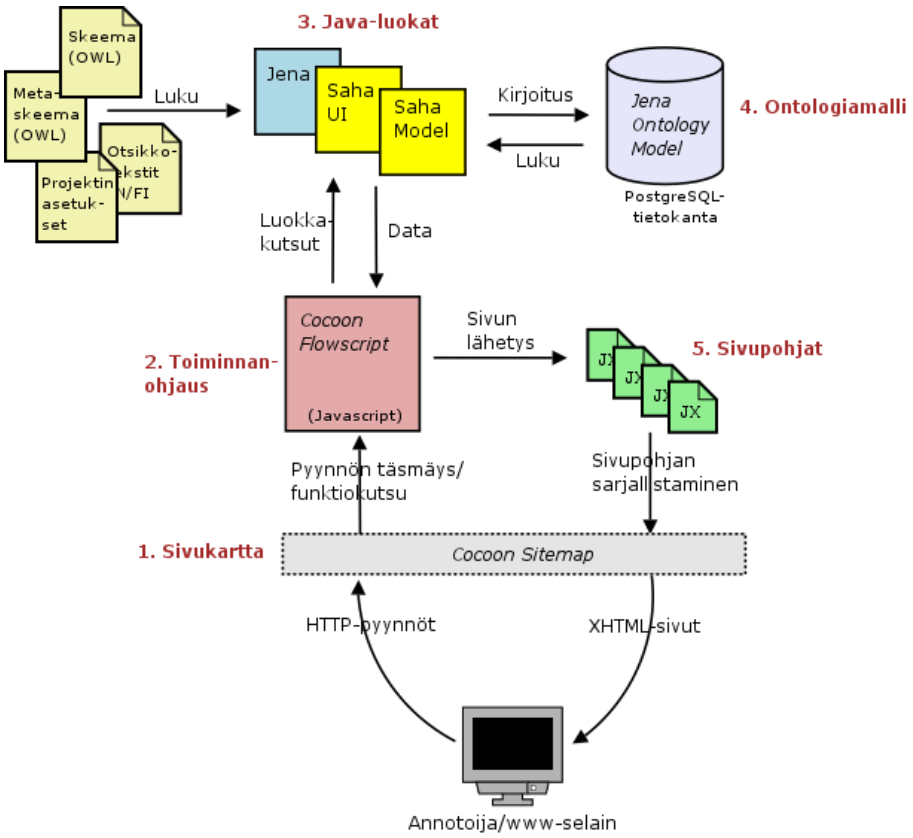
Sahalla suoritettava annotointi perustuu annotaatioskeemaan, joka ladataan Sahaan OWL-muotoisena ontologiana. Sahaan ladattava annotaatioskeema muodostaa *ontologiamallin*, joka tallennetaan tietokantaan. Ontologiamalli, jota Sahassa kutsutaan annotaatioprojektiksi, sisältää annotaatioskeemassa määritellyt luokat ja niiden ominaisuudet sekä kaikki niistä luodut ilmentymät, jotka voivat kuvailla erilaisia resursseja tai olla varsinaisia dokumentteja kuvaavia annotaatioita. Annotoijat pystyvät luomaan annotaatioprojekteihin uusia annotaatioita tai muokkaamaan olemassa olevia ottamalla www-selaimella yhteyden Sahaan. Saha muodostaa annotoijan tekemien valintojen mukaan www-selaimelle lähetettäviä HTML-sivuja, jotka sisältävät annotaatioiden luomiseen ja muokkaamiseen tarkoitettuja lomakkeita. Lomakkeiden avulla annotoija voi syöttää arvoja annotaatioiden ominaisuuksille tai hakea niitä ONKI-palvelimen kautta. Annotaatioita hyödyntävät sovellukset voivat hakea Sahan tietokantaan tallennettuja annotaatioprojekteja RDF/XML-muodosssa HTTP-pyyntöjen avulla.

Sahaan ladattu annotaatioskeema muodostaa annotaatioprojektin, jonka puitteissa tietty joukko Sahan käyttäjiä tuottaa annotaatioita. Yhdelle palvelimelle asennetussa Sahassa voi olla yksi tai useampi annotaatioprojekti ja jokaisella projektilla voi olla yksi tai useampi käyttäjä. Annotaatioprojektien avulla voidaan toisin sanoen tuottaa tietyssä Saha-sovelluksessa samaan aikaan toisistaan riippumatta eri annotaatioskeemoihin perustuvia annotaatioita. Tietyssä annotaatioprojektissa muodostettuja annotaatioita voidaan tarkastella ja muokata vain kyseisen projektin sisällä. Eri projektien annotaatiot voidaan tallentaa joko samaan tai erillisiin tietokantoihin. Annotaatioskeeman lisäksi projektiin liittyy skeeman toimintaa kuvaileva *metaskeema*, sekä projektin yleisiä asetuksia määrittelevä asetustiedosto. Näistä on kerrottu tarkemmin luvuissa 5.5 ja 5.6.1.

5.3 Tekninen toteutus

5.3.1 Yleiskuvaus

Saha on web-sovellus, jonka keskeiset toiminnot on toteutettu Java-ohjelmointikieleen pohjautuvilla vapaan lähdekoodin Apache Cocoon-palvelinohjelmistolla[25] sekä RDF-pohjaisen tiedon käsittelyyn suunnitellulla Jena-ympäristöllä[26]. Edellä mainittujen järjestelmien valintaan vaikuttivat niiden vapaa saatavuus, yleisyys ja soveltuvuus järjestelmälle asetettuihin tavoitteisiin. Sahan kehityksen alkuvaiheessa tutkittiin myös PHP-pohjaisen pOWL-ympäristön (Auer, 2005) soveltuvuutta järjestelmän toteuttamiseen, mutta siitä päätettiin luopua toiminnassa havaittujen puutteiden vuoksi. Cocoonin ja Jenan lisäksi Sahassa hyödynnetään Ajax-tekniikkaa, jonka avulla useimmat Sahan käyttöliittymän toiminnot on toteutettu. Ajaxin avulla voidaan välittää tietoa selaimen ja palvelimen välillä ilman, että selaimessa olevaa sivua täytyy ladata uudelleen. Tekniikka parantaa sivujen käytettävyyttä ja nopeutta. Sahan arkkitehtuuria on havainnollistettu kuvassa 13.



Kuva 13. *Sahan arkkitehtuuri*

[25] http://cocoon.apache.org/

[26] http://jena.sourceforge.net/

Sahan kuvassa 13 esitetty arkkitehtuuri rakentuu seuraavista osista:

1. **Sivukartta**. Sivukartan eli Cocoonin *Sitemapin* tehtävänä on ottaa vastaan annotoijan www-selaimelta saapuvat HTTP-pyynnöt ja pyynnöstä riippuen kutsua niitä vastaavia Javascript-funktioita Cocoonin Flowscriptissä (ks. seuraava kohta). Lisäksi Sitemap huolehtii annotoijan www-selaimelle lähetettävien XHTML-sivujen luonnista sivupohjien perusteella.

2. **Toiminnanohjaus**. Sahan suoritusta ohjaa Cocoonin *Flowscript*, joka on Javascript-sovellusrajapinta. Sahassa se on käytännössä joukko Javascript-funktioita, joita kutsutaan Sahalle tulevien HTTP-pyyntöjen perusteella sivukartasta. Javascript-funktioista kutsutaan edelleen Java-luokkia, jotka käsittelevät annotointiin liittyvän RDF-pohjaisen tiedon. Lisäksi funktioiden avulla lähetetään käyttäjälle muodostettavat XHTML-sivut (Extensible HyperText Markup Language). Flowscriptin avulla Saha toimii tavallisen sovelluksen tapaan, eikä äärellisenä tilakoneena kuten normaalit web-sovellukset. Tämän ansiosta tiettyyn käyttäjään liittyen voidaan tallentaa muuttujia, joiden arvot säilyvät sovelluksen muistissa käyttäjän lähettämien sivupyyntöjen välillä.

3. **Java-luokat**. Annotaatioiden käsittely ja tallennus tapahtuvat Sahassa Java-luokilla. Tähän käytetään Javan Jena-luokkakirjastoa, joka sisältää työkalut RDF-pohjaisen tiedon käsittelyyn. Jenan avulla voidaan tallentaa tietokantaan Sahaan ladattu OWL-muotoinen annotaatioskeema ja sen pohjalta muodostetut annotaatiot. Tietokantaan tallennettu skeema muodostaa ontologiamallin, jonka käsittelyyn Jena tarjoaa tarvittavat rajapinnat. Java-luokkien avulla luetaan myös annotaatioprojektikohtainen asetustiedosto ja annotaatioskeeman toimintaa ohjaava OWL-muotoinen metaskeema.

4. **Ontologiamalli**. Sahassa käytettävät ontologiat, eli annotaatioskeemat tallennetaan PostgreSQL-tietokantaan Jenan ontologiamallina. Jokainen sahaan ladattu annotaatioskeema muodostaa oman ontologiamallinsa ja se tallennetaan tietokantaan erillisenä muista malleista. Sahaa ei ole sidottu tiettyyn tietokantaan, vaan annotaatioprojektikohtaisesti voidaan määritellä tietokanta, johon projekti tallennetaan.

5. **Sivupohjat**. Sahan annotoijalle lähettämät HTML-sivut muodostetaan valmiiksi määriteltyjen sivupohjien perusteella. Sivupohjat on määritelty Cocoonin JXTemplate-kielellä, jonka avulla muodostettaviin sivuihin voidaan sisällyttää tietoa Java- ja Javascript-olioista, jotka välitetään sivupohjiin toiminnanohjauksen (Flowscript) kautta. Sivupohjat muutetaan ennen annotoijalle lähettämistä sivukartassa XHTML-muotoon, jotta ne voidaan esittää käyttäjän www-selaimessa. Sahassa on määritelty omat sivupohjat käyttöliittymän eri osille, kuten sisäänkirjautumissivulle, annotointisivulle jne.

5.3.2 Java-luokat

Annotaatioiden muokkaamiseen ja tallentamiseen liittyvät toiminnot on toteutettu Sahassa seuraavilla viidellä Java-luokalla:

1. **SahaModel**. Luokka muodostaa yhteyden tietokantaan ja tarjoaa metodit sinne tallennettujen Jena-ontologiamallien muokkaamiseen. Luokka sisältää metodit mm. ontologiamallissa olevien luokkien instantiointiin, ominaisuuksien arvojen määrittelyyn, muokkaamiseen ja poistamiseen sekä ilmentymien hakemiseen niille määriteltyjen ominaisuuksien perusteella.

2. **SahaUi**. Luokka toimii SahaModel-luokan ja Sahan toiminnanohjauksen (flowscript) välissä. Se muuttaa Jenan ontologiamallista saatavat tiedot sellaiseen muotoon, että ne voidaan sisällyttää Cocoonin JXTemplate-sivupohjien avulla muodostettaviin HTML-sivuihin.

3. **SahaSettings**. Luokan avulla luetaan muistiin annotaatioprojektiin liittyvistä asetustiedostoista projektin yleiset asetukset, metaskeema sekä kielikohtaiset käyttöliittymän otsikkotekstit. Luokka tarjoaa metodit asetusten hakemiseen muistista.

4. **SahaTemporaryStore**. Luokkaa käytetään tallentamaan muokattavan annotaation tiedot muokkauksen alkaessa. Mikäli annotaatioon tehdyt muutokset halutaan peruttaa, saadaan alkuperäiset arvot palautettua luokan avulla.

5. **SahaManager**. Luokkaa käytetään annotaatioprojektien hallintaan. Luokan avulla voidaan lukea tekstitiedostosta RDF/OWL-muotoinen ontologia (annotaatioskeema) ja luoda siitä uusi ontologiamalli. Luokka tarjoaa metodin mallien sarjallistamiseen RDF/XML-muotoon ja kirjoittamiseen tekstitiedostoon.

5.3.3 Annotaatioskeema

Sahassa annotaatioiden rakenne määritellään OWL-kielellä kuvatussa annotaatioskeemassa. Skeemaa käytetään annotaatioiden rakenteen kuvailun lisäksi Sahan käyttöliittymän muodostamisessa. Annotaatioskeema voidaan ladata Sahaan tekstitiedostosta, johon se on kirjoitettu RDF/XML-muodossa. Liitteessä 1 on esimerkki tällaisesta annotaatioskeemasta. Kun Sahaan luodaan uusi annotaatioprojekti, annotaatioskeema luetaan tiedostosta ja sen perusteella luodaan Jenan ontologiamalli, joka tallennetaan tietokantaan. Tämän jälkeen alkuperäistä skeematiedostoa ei enää käytetä, vaan skeeman pohjalta muodostettavat annotaatiot tallennetaan tietokantaan. Tietokannassa oleva ontologiamalli voidaan myöhemmin sarjallistaa RDF/XML-muotoon ja tallentaa takaisin tekstitiedostoon.

5.4 Käyttöliittymä

5.4.1 Yleistä

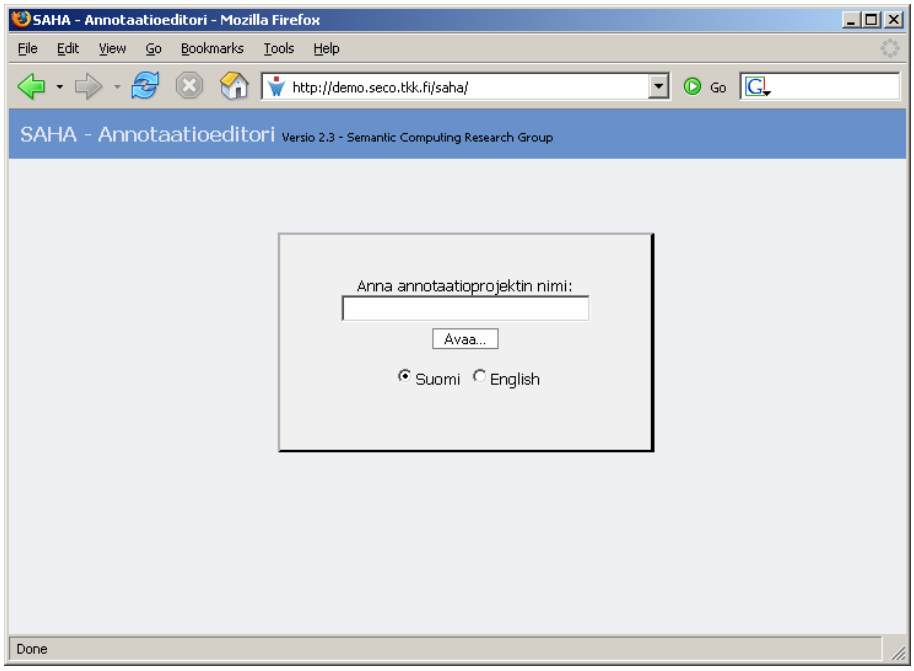
Sahan käyttöliittymä muodostuu palvelimella dynaamisesti luotavista HTML-sivuista. Käyttöliittymä muodostetaan käyttäjän valitseman annotaatioprojektin ja siihen liittyvän annotaatioskeeman sekä sen toimintaa projektissa kuvaavan metaskeeman pohjalta. Käyttöliittymän tarkoituksena on tarjota käyttäjälle havainnollinen näkymä tiettyyn annotaatioskeemaan liittyvistä luokista, ominaisuuksista ja instansseista sekä mahdollisuus hyödyntää niitä annotaatioiden muodostamiseksi. Käyttöliittymän suunnittelun lähtökohtana oli toteuttaa mahdollisimman yksinkertainen näkymä annotoinnin taustalla oleviin ontologioihin, jolloin ontologisesti kuvatun tiedon tuottaminen olisi helppoa mahdollisimman laajalle käyttäjäjoukolle.

Koska Sahan käyttöliittymästä on pyritty tekemään mahdollisimman yksinkertainen, on sen ominaisuuksista rajattu pois monimutkaisemmat ontologioiden muokkaamiseen liittyvät toiminnot. Tällainen rajaus oli perusteltua, koska annotoijan tehtävät voidaan usein rajata olemassa olevien luokkien instantiointiin ja niiden ominaisuuksien arvojen määrittämiseen. Yksinkertaisen käyttöliittymän lisäksi suunnittelun tavoitteena oli myös se, että annotoijan ei tarvitsisi välttämättä huolehtia sovelluksen käyttöönotosta, ylläpidosta ja annotaatioiden tallentamisesta. Nämä tavoitteet pyrittiin saavuttamaan tekemällä Sahasta web-sovellus, joka toimii käyttäjän koneen sijasta palvelimella. Sahan käyttö ei vaadi annotoijalta muuta kuin sopivan www-selaimen ja Internet-yhteyden, annotaatioiden muodostaminen ja tallennus tapahtuvat palvelimella. Annotaatioiden keskitetty tallentaminen on lisäksi tehokas tapa jakaa annotaatioita muiden annotoijien kesken, mitä voidaan pitää keskeisenä vaatimuksena semanttisesti yhteensopivien annotaatioiden tuottamisessa.

Sahan käyttöliittymä voidaan jakaa kahteen perusosaan, joilla kummallakin on oma tehtävänsä annotointiprosessissa. Ensimmäistä osaa voidaan kutsua *luokkasivuksi* (kuva 15), koska se tarjoaa näkymän annotaatioskeemassa määriteltyihin annotaatioluokkiin, joita käytetään pohjana uusien annotaatioiden luonnissa. Toista osaa voidaan taas nimittää *annotaatiosivuksi* (kuva 17), koska se muodostaa näkymän tiettyyn annotaatioon liittyviin ominaisuuksiin ja niille määriteltyihin arvoihin sekä tarjoaa mahdollisuuden luoda uusia arvoja tai muokata olemassa olevia. Seuraavissa aliluvuissa on selostettu Sahan toimintaa käyttöliittymän eri osien kautta. Sahalla suoritettavan annotointiprosessin kokonaiskuvaus on esitetty UML-kaavioina luvussa 5.8.

5.4.2 Sisäänkirjautumissivu

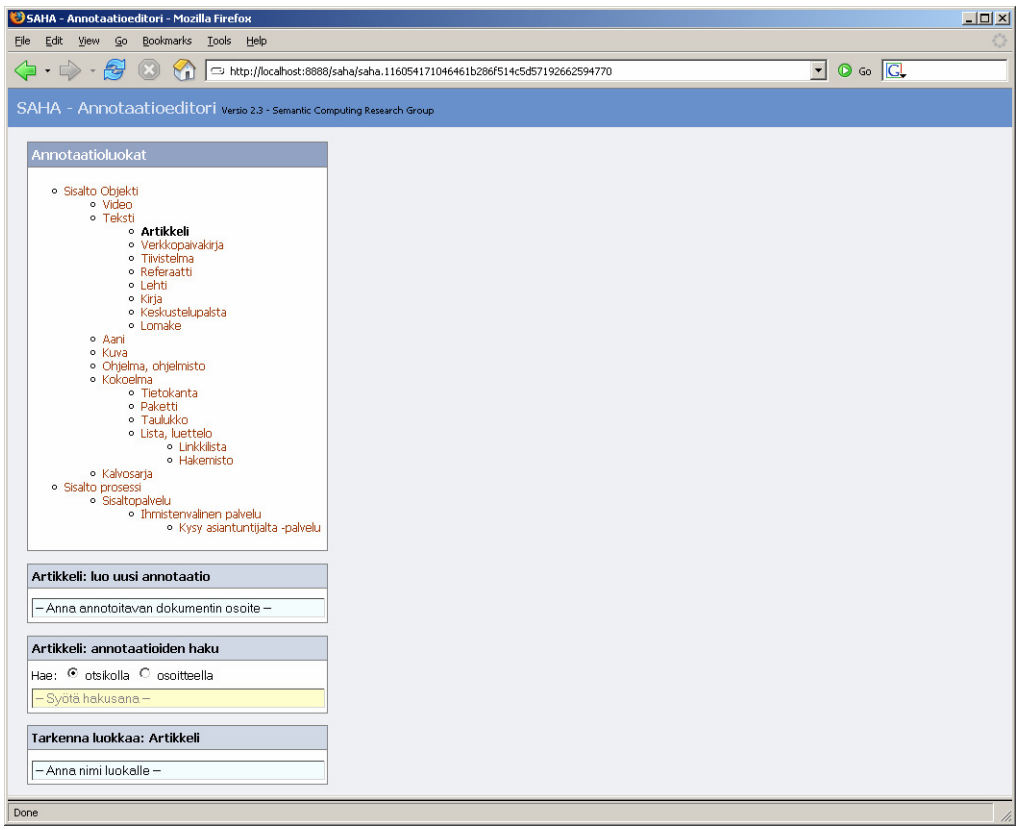
Alkaessaan käyttää sahaa, annotoija avaa aluksi Sahan sisäänkirjautumissivun (kuva 14). Sisäänkirjautumissivulla käyttäjä syöttää haluamansa annotaatioprojektin nimen ja valitsee haluamansa kielen. Sisäänkirjautumissivulta siirrytään luokkasivulle.



Kuva 14. *Sahan sisäänkirjautumissivu*

5.4.3 Luokkasivu

Sahan luokkasivulla (kuva 15) annotoija näkee valitsemansa annotaatioprojektin annotaatioluokkien hierarkian sekä sen alapuolella kentät, joiden avulla voidaan hakea aikaisemmin luotuja annotaatioita, luoda uusi annotaatio ja tarkentaa luokkahierarkiaa luomalla siihen uusia aliluokkia. Kenttien toiminnot kohdistuvat aina siihen luokkaan, joka on valittu luokkahierarkiasta. Kuvassa 15 hierarkiasta on valittu luokka ”Artikkeli”, jolloin esimerkiksi uusi annotaatio luotaisiin käyttäen kyseistä luokkaa. Annotaatioita haettaessa haut kohdistuvat aina valitun luokan sekä kaikkien sen aliluokkien ilmentymiin. Annotoija voi valita hakeeko annotaatioita annotoidun dokumentin osoitteella vai annotaatioiden otsikoilla. Ensimmäinen hakutapa on hyödyllinen silloin, kun halutaan nähdä kaikki tiettyyn dokumenttiin liittyvät annotaatiot tai kun halutaan muokata johonkin tiettyyn dokumenttiin liitettyä annotaatiota. Jälkimmäinen soveltuu taas tilanteeseen, jossa ei tiedetä halutun dokumentin osoitetta, mutta halutaan muokata siihen liittyviä annotaatioita.



Kuva 15. *Sahan luokkasivu*

Uuden annotaation luonti tapahtuu syöttämällä annotoitavan dokumentin osoite (URL) käyttöliittymässä olevaan kenttään. Mikäli annotaatioprojektissa käytetään dokumentteja luokittelevaa annotointia, tarkistaa Saha ennen uuden annotaation luontia, että syötettyä dokumenttia ei ole annotoitu jo aikaisemmin. Tarkistus tehdään, koska Sahassa voidaan luokittelevassa annotoinnissa annotoida yksi dokumentti vain yhdellä luokalla, eli toisin sanoen yksi dokumentti ei voi olla useamman annotaatioskeemassa määritellyn luokan ilmentymä. Mikäli dokumentti on annotoitu jo aikaisemmin, ei uutta annotaatiota luoda, vaan avataan aikaisemmin dokumentista muodostettu annotaatio. Mikäli annotaatioprojektissa ei käytetä dokumenttien luokittelua, vaan annotaatiot liitetään dokumenttiin ominaisuuden avulla, ei edellä kuvattua tarkistusta tehdä. Tällaisessa annotoinnissa yhteen dokumenttiin voidaan liittää rajoittamaton määrä eri luokista muodostettuja annotaatioita.

Kuvassa 16 on esitetty annotaatioiden hakukenttä, joka on osa kuvassa 15 esitettyä luokkasivua. Siinä annotoija hakee luokkaan ”Artikkeli” kuuluvia annotaatioita hakusanalla ”seco”. Koska annotoija on valinnut haun kohdistumaan annotaatioiden otsikoihin, verrataan hakusanaa kaikkiin luokan ”Artikkeli” ilmentymiin, joiden `rdfs:label`-otsikossa esiintyy käytetty hakusana. Haku on palauttanut tulokseksi yhden annotaation, jonka otsikko on ”Secon kotisivu”. Hakutulos näytetään haun päätyttyä hakukentän alapuolelle avautuvassa valikossa. Valikossa on myös linkki, jota klikkaamalla voidaan listata kaikki luokkaan liittyvät annotaatiot sekä linkki, josta hakutulosvalikko voidaan sulkea. Klikkaamalla hakutuloksissa olevaa annotaatiota, avataan kyseinen annotaatio muokattavaksi. Mikäli annotoija olisi valinnut osoitteen perusteella tehtävän

haun, olisi hakusanana käytetty sen dokumentin osoitetta (URL), jonka annotaatio halutaan löytää.

Kuva 16. *Annotaatiohaku*

Annotaatioiden haku on toteutettu siten, että haku käynnistyy taustalla välittömästi, kun annotoija syöttää merkin hakukenttään. Toteutus muistuttaa Googlen Suggest-hakua[27] sekä Hyvösen ja Mäkelän (2006) *automaattista semanttista täydentämistä* (semantic autocompletion). Haku tarkentuu sitä mukaa kun käyttäjä syöttää uusia merkkejä, koska jokaisen merkin jälkeen käynnistetään uusi haku, joka ohittaa edellisen haun. Hakutulokset tulevat näkyviin hakukentän alle ilmestyvään valikkoon. Mikäli haulla löydetään annotaatioita, niiden otsikot näkyvät valikossa ja ne voidaan avata otsikkoa klikkaamalla.

Mikäli annotaatioprojektissa on sallittu luokkahierarkian laajentaminen uusilla aliluokilla, on uuden annotaation luontikentän alapuolella kenttä uuden aliluokan nimen määrittämiseen (kuva 15). Aliluokan luonti tapahtuu antamalla nimi aliluokalle, jolloin sen niminen aliluokka luodaan luokkahierarkiasta aiemmin valitulle luokalle. Uusien aliluokkien luonnista on kerrottu tarkemmin luvussa 5.4.7.

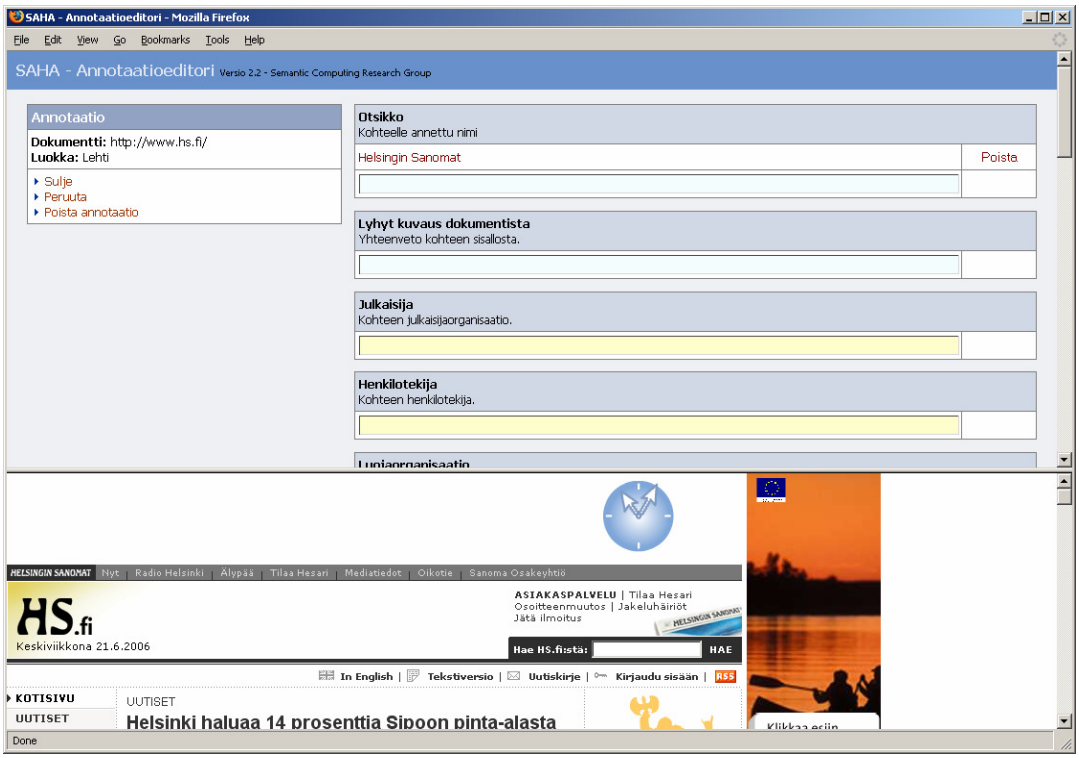
5.4.4 Annotaatiosivu

Annotaatiosivulla (kuva 17) esitetään tiettyyn dokumenttiin liittyvä annotaatio sekä annotoitava dokumentti, mikäli käytettävä www-selain pystyy näyttämään sen[28]. Sivun vasemmassa yläkulmassa näkyvät annotoitavan dokumentin URL-osoite sekä valitun annotaatioluokan nimi. Niiden alla on valikko, josta voidaan sulkea annotaatio, peruuttaa annotaatioon sen avaamisen jälkeen tehdyt muutokset sekä poistaa annotaatio. Oikealla puolella näkyvät annotaatioon liittyvän luokan ominaisuudet, niille määritellyt arvot sekä kentät uusien arvojen syöttämiseen. Annotaatiota ei tarvitse erikseen tallentaa sen jälkeen kun siihen on tehty muutoksia, koska kaikki muutokset (ominaisuuksien arvojen lisääminen ja poisto) tallennetaan pysyvästi niiden tekohetkellä.

---

[27] http://labs.google.com/suggest/

[28] Sahalla voidaan annotoida mitä tahansa resursseja, joihin voidaan osoittaa URI-osoitteella. Käyttöliittymässä voidaan kuitenkin näyttää vain sellaisia dokumentteja, joiden näyttämistä käytettävä www-selain tukee. Mikäli selain ei tue annotoitavan dokumentin formaattia, näytetään tyhjä sivu.



Kuva 17. *Sahan annotaatiosivu*

Annotaatioon liittyvät ominaisuudet näkyvät annotaatiosivulla omissa alueissaan. Jokaisesta ominaisuudesta näytetään ensin ominaisuuden otsikko (rdfs:label) sekä mahdollinen kuvaus (rdfs:comment) ja niiden jälkeen lista arvoista, joita ominaisuudelle on määritelty. Kuvassa 18 on esitetty osa kuvan 17 annotaatiosivun lomakkeesta. Siinä literaaliominaisuudelle ”Otsikko”, jonka kuvauksena on ”Kohteelle annettu nimi”, on syötetty arvo ”Helsingin Sanomat”.

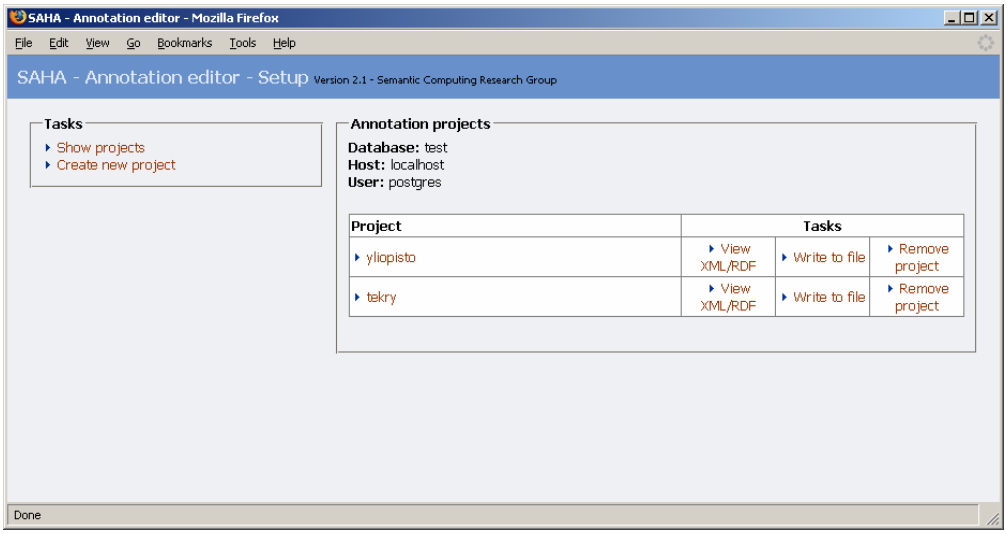
Kuva 18. *Literaaliominaisuuden ”Otsikko” syöttökenttä*

Skeemasta riippuen jokaiselle ominaisuudelle voidaan sallia yksi tai useampi arvo. Jokaista arvoa voidaan muokata klikkaamalla arvoa, jolloin ominaisuuden tyypistä riippuen näytetään joko arvon muokkauskenttä (literaaliarvot) tai avataan arvon muokkausikkuna (objektiarvot). Arvotaulukon oikeassa reunassa on jokaisen arvon kohdalla ”Poista”-painike, jota painamalla kyseinen arvo voidaan poistaa. Mikäli poistettava arvo liittyy objektiominaisuuteen, ei arvon poistaminen poista arvona olevaa ilmentymää tai käsitettä, vaan ainoastaan annotaatiossa olevan viittauksen siihen. Arvotaulukon alapuolella on kenttä, jonka avulla voidaan määritellä uusi arvo ominaisuudelle. Mikäli kyseessä on

literaaliominaisuus, tallennetaan kenttään syötetty merkkijono sellaisenaan ominaisuuden arvoksi. Jos taas kyseessä on objektiominaisuus, toimii kenttä hakukenttänä joko ilmentymäarvoille tai ONKI-palvelulle. Objektiominaisuuksien määrittämisestä on kerrottu tarkemmin luvussa 5.4.6. Annotaatiosivulla objektiominaisuuksien syöttö- ja hakukenttä näkyy keltaisella pohjalla, kun taas literaaliominaisuuksien syöttökenttä (kuva 18) on vaaleansininen.

5.4.5 Hallintasivu

Annotaatioprojekteja voidaan hallita Sahan hallintasivuilta, joka tarjoaa näkymän kaikkiin järjestelmään määriteltyihin annotaatioprojekteihin sekä lomakkeen, jonka avulla voidaan luoda uusi annotaatioprojekti. Hallintasivu on esitetty kuvassa 19. Annotaatioprojektit näkyvät hallintasivulla taulukossa, jonka vasemmassa reunassa näkyvät projektien nimet ja kunkin projektin oikealla puolella toiminnot, joita niihin voidaan kohdistaa. Toimintoja ovat projektien katselu ja tallentaminen tiedostoon RDF/XML-muodossa sekä projektien poistaminen. Projekti voidaan avata klikkaamalla sen nimeä, jolloin siirrytään projektin luokkasivulle.



Kuva 19. *Sahan hallintasivu*

5.4.6 Objektiominaisuuksien arvojen määrittäminen

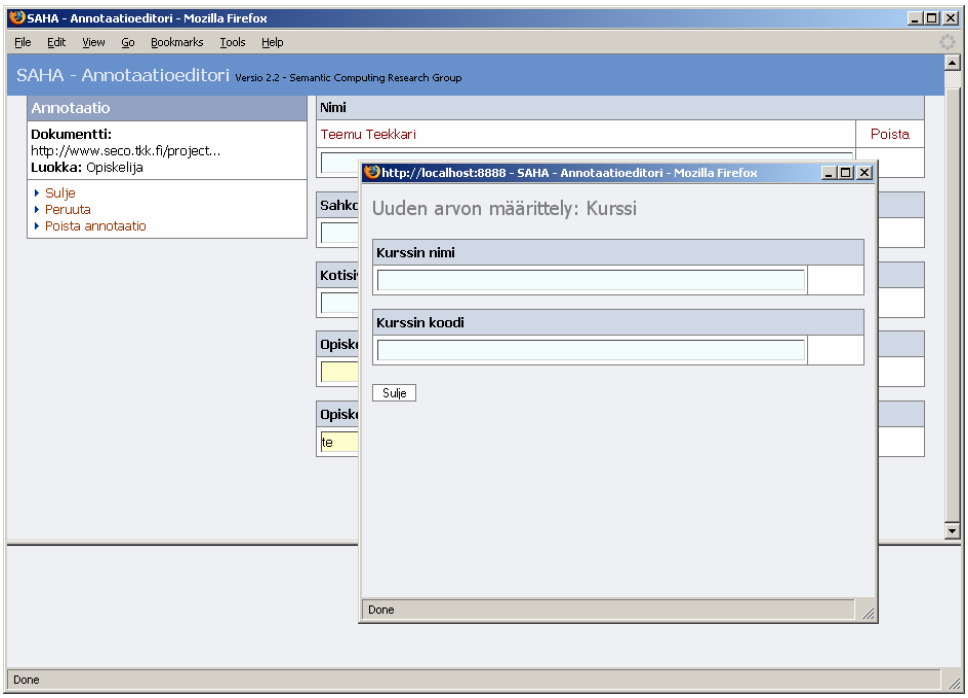
Objektityyppinen ominaisuus voi Sahassa saada arvokseen joko annotaatioskeeman luokista muodostettuja ilmentymiä tai referenssiontologioissa määriteltyjä käsitteitä. Määrittäessään Sahassa objektityyppiselle ominaisuudelle ilmentymäarvoa, annotoijalla on mahdollisuus valita ominaisuuden arvoksi joko olemassa olevan ilmentymä tai luoda määrittelyhetkellä uusi. Ilmentymätyyppisen arvon määrittely on toteutettu Sahassa siten, että annotoijaa ohjataan ensisijaisesti käyttämään annotaatioprojektissa aiemmin määriteltyjä ilmentymiä ja luomaan uusia vain siinä tapauksessa, että sopivaa ilmentymää ei löydy. Toimimalla näin pyritään varmistamaan, että tiettyä resurssia, kuten henkilöä edustaisi ainoastaan yksi ilmentymä, johon eri annotaatioista viitataan.

Ilmentymätyyppisen ominaisuuden arvon määrittämiseen liittyvät vaiheet on esitetty UML-kaaviona luvussa 5.8.

Ilmentymäarvon määrittely tapahtuu siten, että annotoija syöttää Sahan annotaatiosivulla objektiarvoisen ominaisuuden määrittelykenttään hakusanan, joka liittyy jollain tavalla määriteltävään arvoon. Jos ollaan määrittelemässä arvoksi esimerkiksi johonkin henkilöön viittaavaa ilmentymää, hakusanana voi toimia esimerkiksi henkilön nimi. Hakusanan syöttäminen käynnistää automaattisesti taustalla tapahtuvan haun, joka kohdistuu kaikkiin sellaisiin annotaatioprojektissa määriteltyihin ilmentymiin, jotka tyyppinsä perusteella sopivat määriteltävän ominaisuuden arvoksi. Haku on teknisesti toteutettu vastaavalla tavalla, kuin luokkasivun annotaatiohaku (ks. luku 5.4.3). Ilmentymiä haettaessa hakusana kohdistetaan niihin ilmentymiin, joiden tyyppi, eli luokka on määritelty ominaisuuden arvoalueessa (rdfs:range). Hakusanaa verrataan arvoalueeseen kuuluvien ilmentymien rdfs:label-otsikoihin. Mikäli käyttäjän syöttämällä hakusanalla löydetään ilmentymiä, näytetään hakutulokset hakukentän alapuolelle haun päätyttyä aukeavassa valikossa, joka on esitetty kuvassa 20 (kuva esittää osaa Sahan annotaatiosivusta). Kuvassa annotoija on hakenut ”Opiskelee kurssilla”-ominaisuuden arvoksi ilmentymiä, joissa esiintyy merkkijono ”as-75.1”. Annotaatioskeemassa on määritelty ominaisuuden arvoalueeksi luokka ”Kurssi”. Saha on palauttanut hakutuloksena kaksi ”Kurssi”-ilmentymää, jotka käyttäjä voi halutessaan valita ominaisuuden arvoksi. Mikäli mikään löytyneistä ilmentymistä ei vastaa käyttäjän etsimää tai yhtään ilmentymää ei löydy, voi käyttäjä listata kaikki ominaisuuden arvoksi sopivat ilmentymät hakutulosvalikossa olevaa ”Hae kaikki arvot”-linkkiä klikkaamalla tai vaihtoehtoisesti luoda uuden ilmentymän klikkaamalla linkkiä ”Määrittele uusi arvo”.

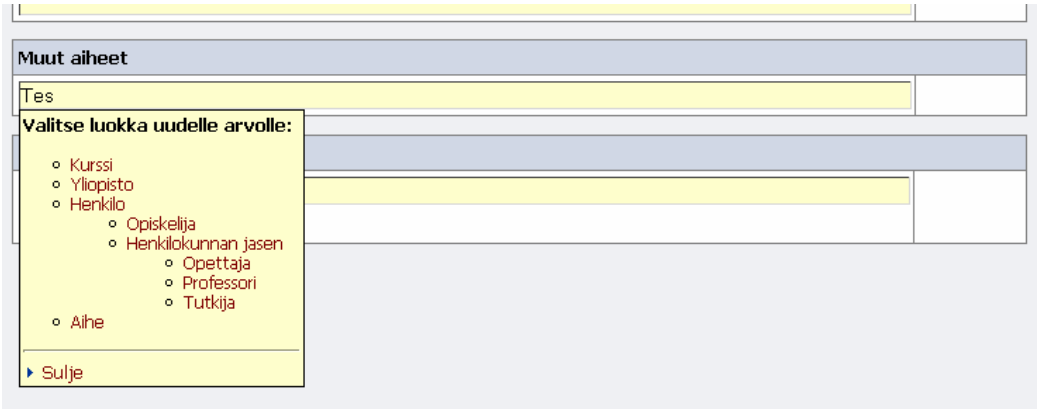
Kuva 20. *Ilmentymähaku*

Ilmentymän luonti tapahtuu uudessa selainikkunassa, joka avataan varsinaiselta annotaatiosivulta. Ilmentymän luonti-ikkunan toiminnallisuus vastaa annotaatiosivun toiminnallisuutta. Kuvassa 21 on esitetty annotaatiosivu ja sen päälle avattu ilmentymän luonti-ikkuna, jonka avulla voidaan määritellä uusi ilmentymä luokasta ”Kurssi”. Luokalla on kaksi ominaisuutta, joiden arvot käyttäjä syöttää vastaavasti kuin syöttäisi arvoja annotaatiosivulla. Kun halutut ominaisuudet on määritelty, suljetaan ikkuna ja tämän jälkeen luotu ilmentymä on alkuperäisen annotaation arvona. Mikäli luotavalla ilmentymällä on objektiominaisuuksia, määritellään ne samalla tavalla kuin edellä on kuvattu. Mikäli tässä tilanteessa halutaan edelleen luoda uusi ilmentymä, avataan nykyisestä ilmentymän luonti-ikkunasta edelleen uusi luonti-ikkuna ja määritellään siinä uuden ilmentymän ominaisuudet.



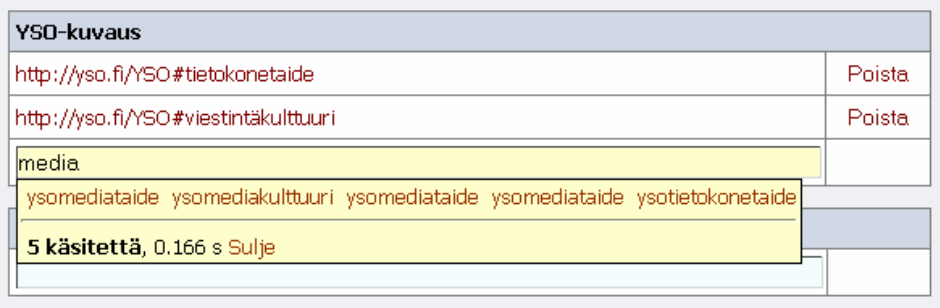
Kuva 21. *Ilmentymän luonti-ikkuna*

Mikäli objektityyppiselle ominaisuudelle ei ole määritelty annotaatioskeemassa arvoaluetta (rdfs:range), voidaan ominaisuuden arvoksi valita mikä tahansa annotaatioskeeman luokan ilmentymä. Määrittäessään uutta ilmentymää tällaisen ominaisuuden arvoksi, käyttäjän tulee ensin valita luokka, josta haluaa muodostaa ilmentymän ja vasta tämän jälkeen avataan uuden ilmentymän luonti-ikkuna. Luokan valinta suoritetaan myös siinä tapauksessa, kun jonkin ominaisuuden arvoalueeseen kuuluu useampi kuin yksi luokka. Luokan valinta tapahtuu ominaisuuden määrittelyssä käytettävän hakukentän alapuolelle avautuvassa valikossa (kuva 22), jossa näytetään ne annotaatioskeeman luokat, joista arvoksi tuleva ilmentymä voidaan luoda. Valikko avataan, kun annotoija valitsee hakutulosvalikosta (kuva 20) uuden ilmentymän luonnin. Mikäli ominaisuuden arvoaluetta ei ole määritelty, näytetään valikossa kaikki annotaatioskeeman luokat. Jos taas arvoalue on määritelty ja siihen kuuluu useampi kuin yksi luokka, näytetään luokkahierarkiassa kaikki arvoalueeseen kuuluvat luokat.



Kuva 22. *Uuden ilmentymän luonti, luokan valinta*

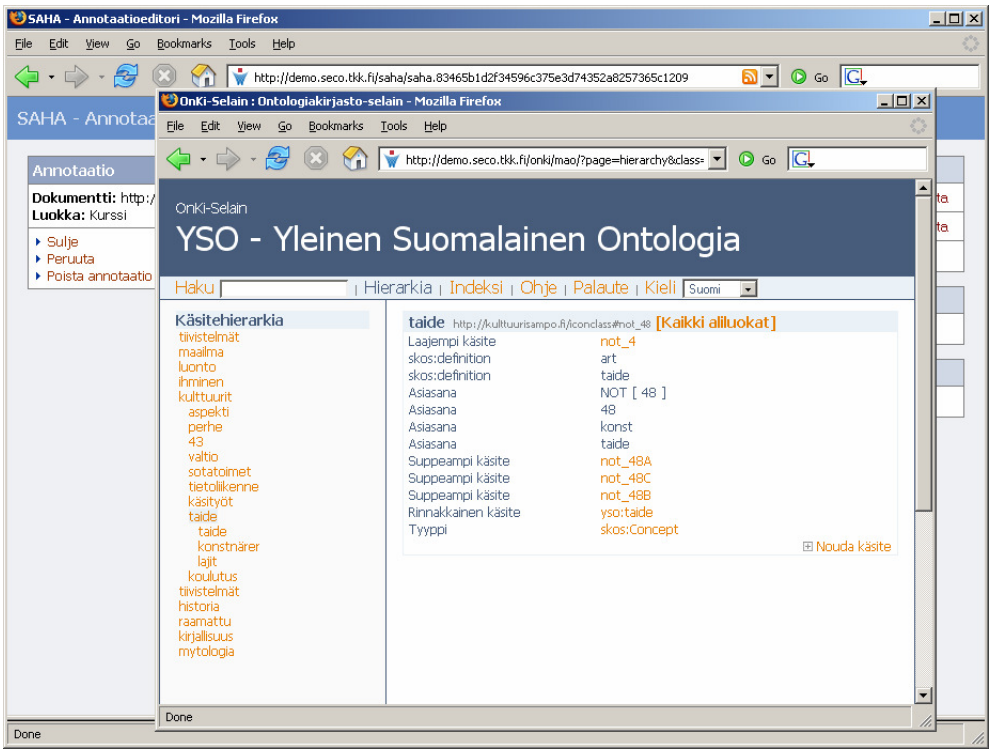
Objektityyppinen ominaisuus voi saada arvokseen annotaatioskeeman pohjalta muodostettujen ilmentymien lisäksi myös jossain skeeman ulkopuolisessa referenssiontologiassa, kuten YSO:ssa määriteltyjä resursseja. Tällaiset arvot syötetään Sahassa ONKI-palvelun avulla. Jokaiselle ominaisuudelle voidaan määritellä annotaatioprojektin metaskeemassa tietty ONKI-palvelin, josta kyseisen ominaisuuden arvo haetaan. Tämä on tarpeen, koska eri ONKI-palvelimet voivat sisältää eri ontologioita. Sahan käyttöliittymä tarjoaa kaksi eri vaihtoehtoa määritellä ominaisuuden arvo ONKI:n avulla. Ensimmäinen tapa vastaa edellä kuvattua ilmentymähakua. Siinä käyttäjä syöttää ominaisuuteen liittyvään hakukenttään hakusanan, joka välitetään taustalla ONKI-palvelimelle. ONKI-palvelin palauttaa hakutuloksena ne referenssiontologian käsitteet, jotka vastaavat hakusanaa. Käsitteet esitetään hakusanakentän alapuolelle ilmestyvässä hakutulosvalikossa, josta käyttäjä voi klikkaamalla valita niitä ominaisuuden arvoksi. Kuvassa 23 on esitetty osa annotaatiosivun lomakkeesta. Siinä annotoija on määrittämässä arvoa objektiominaisuudelle ”YSO-kuvaus”, jolle on jo aiemmin määritelty kaksi arvoa. Annotoija on syöttänyt hakukenttään hakusanan ”media”, jonka perusteella ONKI-palvelin on löytänyt viisi hakusanaa vastaavaa käsitettä, jotka näkyvät hakukentän alla olevassa valikossa[29]. Annotoija voi sulkea valikon klikkaamalla linkkiä ”Sulje”.



Kuva 23. *Käsitteen haku ONKI-palvelimelta*

Toisessa ONKI-palvelun käyttötavassa annotoija valitsee haluamansa referenssiontologian käsitteen käyttämällä ONKI-*selainta*. Tämä hakutapa sopii esimerkiksi tilanteeseen, jossa annotoija ei löydä syöttämällään hakusanalla haluamaansa käsitettä ja haluaa etsiä sen referenssiontologian luokkahierarkiaa tutkimalla. ONKI-selain voidaan avata hakusanakentän vieressä olevasta painikkeesta Sahan selainikkunan päälle aukeavaan uuteen selainikkunaan (kuva 24). Kun annotoija on löytänyt ONKI:sta haluamansa käsitteen, sen voi siirtää Sahassa ominaisuuden arvoksi klikkaamalla käsitteen vieressä olevaa ”Nouda käsite”-linkkiä. Tällöin ONKI-selain suljetaan ja valittu käsite välitetään Sahassa ominaisuuden arvoksi. Kuvassa 24 esitetään tilanne, jossa annotoija on avannut Sahan annotaatiosivulta ONKI-selaimen ja valinnut siellä olevasta YSO-ontologiasta käsitteen ”taide”. Klikkaamalla ”Nouda käsite”-linkkiä, käsitteen ”taide” URI asetetaan Sahassa määriteltävän ominaisuuden arvoksi..

[29] Kuvassa 23 esitetyssä tilanteessa on käytössä ONKI-palvelimen kehitysversio. Tämän johdosta käsitteiden otsikot eivät näy hakutulosvalikossa täydellisinä, vaan sisältävät mm. ylimääräisen ontologiaan viittaavan etuliitteen ”yso”.



Kuva 24. *Käsitteen haku ONKI-selaimella*

5.4.7 Uusien aliluokkien luonti annotaatioskeemaan

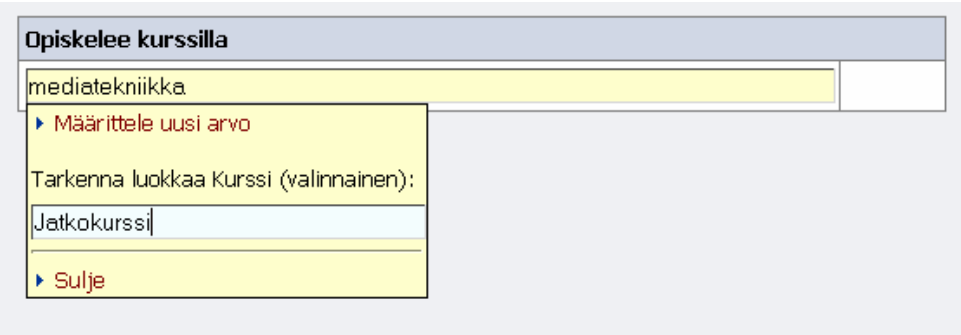
Annotaatioskeeman suunnitteluvaiheessa ei välttämättä tarkasti tiedetä tai osata ennustaa minkä tyyppisiä asioita skeeman avulla tullaan annotoimaan. Tästä johtuen skeeman luokkahierarkia ei välttämättä kaikissa tilanteissa määrittele annotoitavan resurssin tyyppiä annotoijan haluamalla tarkkuudella. Yksinkertaisena esimerkkinä voidaan ajatella tilanne, jossa annotoidaan eläimiä käsitteleviä dokumentteja tarkoitusta varten laaditulla annotaatioskeemalla. Jos annotoitavassa dokumentissa käsitellään esimerkiksi pöllöjä ja annotaatioskeema sisältää ainoastaan luokan ”Linnut”, ei skeeman luokkahierarkian avulla voida tarkasti ilmaista, minkälaisista linnuista on kyse. Tällaisessa tapauksessa annotoijalla voi olla tarve tarkentaa määriteltyä luokkahierarkiaa.

Sahassa annotoijien on mahdollista luoda annotaatioskeemassa määritellyille luokille uusia aliluokkia kuvatakseen tarkemmin annotoitavia resursseja. Käytäntö muistuttaa Annoteassa (Kahan ym. 2001) esitettyä mallia, jossa käyttäjät pystyvät määrittelemään annotaation tyypin luomalla uusia aliluokkia järjestelmän yleisille annotaatioluokalle. Kun resurssi on annotoitu alkuperäiseen luokkahierarkiaan lisätyllä uudella aliluokalla, voidaan annotaatioiden perusteella suoritettavia hakuja kohdistaa aiempaa tarkemmin juuri tietyn tyyppisiin resursseihin. Kun annotoija pystyy edellä esitetyssä lintudokumenttien annotointiesimerkissä luomaan luokalle ”Linnut” uuden aliluokan ”Pöllöt”, voidaan myöhemmin hakea annotaatioiden perusteella kaikki ne dokumentit, jotka liittyvät pöllöihin ja jättää muut linnut haun ulkopuolelle.

Saha ei tue uusien ominaisuuksien määrittämistä luoduille aliluokille, mutta ne perivät kaikki yläluokilleen määritellyt ominaisuudet. Tämän johdosta annotoijien luomia aliluokkia on jälkikäteen tarvittaessa helppoa yhdistellä tai poistaa, koska uusien aliluokkien pohjalta luoduilla ilmentymillä on samat ominaisuudet kuin niiden yläluokilla. Yhdistäminen tulee kyseeseen esimerkiksi silloin, kun luokkahierarkiaan on luotu kaksi rinnakkaista samaan asiaan viittaavaa aliluokkaa. Poistaminen voidaan taas toteuttaa silloin, kun jokin annotaatioskeemaan luotu aliluokka katsotaan merkitykseltään tarpeettomaksi, tai jollain tapaa virheelliseksi. Jos jokin uusi aliluokka halutaan poistaa ja muuttaa siitä muodostettujen ilmentymien tyyppi alkuperäiseksi yläluokaksi, ei tyyppimuunnoksessa menetetä muuta tietoa kuin aliluokalle määritelty nimi.

Uusien aliluokkien luonti voidaan annotaatioprojektikohtaisesti joko sallia tai estää. Mikäli aliluokkien luonti on sallittu, voi annotoija määritellä uusia aliluokkia sekä annotaatio- että referenssiluokille. Käytännössä tämä tarkoittaa sitä, että uuden luokan luontimahdollisuus tarjotaan kaikissa niissä tilanteissa, joissa jostain annotaatioskeeman luokasta luodaan uusi ilmentymä. Tällaisia tilanteita ovat uuden annotaation luonti Sahan luokkasivulla sekä objektiominaisuuden arvon määrittäminen annotaatiosivulla. Annotaatioluokille voidaan luoda uusia aliluokkia luokkasivulla, joka on esitetty kuvassa 15. Siinä luokkahierarkian alla näkyy kenttä, jonka yläpuolella on otsikko ”Tarkenna luokkaa: Artikkeli”. Kun kenttään syötetään uuden aliluokan nimi, luodaan annotaatioskeemaan uusi aliluokka luokalle ”Artikkeli”. Tämän jälkeen annotoija voi muodostaa uuden annotaation käyttäen luomaansa aliluokkaa.

Annotointisivulla tapahtuva uuden aliluokan määrittäminen on esitetty kuvassa 25, jossa näkyy osa annotaatiosivun lomakkeesta. Siinä annotoija on määrittämässä arvoa objektityyppiselle ominaisuudelle ”Opiskelee kurssilla”, joka saa arvoikseen luokan ”Kurssi” ilmentymiä. Annotoija on aluksi etsinyt aiemmin luotuja ”Kurssi”-ilmentymiä hakusanalla ”mediatekniikka” ja koska yhtään ilmentymää ei ole löytynyt, valinnut uuden ilmentymän luonnin (vrt. kuva 20, ilmentymähaku). Tässä vaiheessa annotoijalla on mahdollisuus luoda uusi aliluokka luokalle ”Kurssi” kirjoittamalla uuden aliluokan nimi sivulla näkyvään kenttään. Annotoija on määritellyt uuden aliluokan, jonka nimi on ”Jatkokurssi”. Klikatessaan seuraavaksi ”Määrittele uusi arvo”-linkkiä, annotoija siirtyy määrittelemään luokan ”Jatkokurssi” ominaisuuksia avautuvassa ilmentymän luonti-ikkunassa.



Kuva 25. *Skeeman luokkahierarkian tarkentaminen uudella aliluokalla*

5.5 Metaskeema

5.5.1 Yleistä

Annotaatioskeemassa ei yleensä ole tarkoituksenmukaista tai välttämättä edes mahdollista kuvailla sitä, miten skeemaa tulisi käyttää annotaatiojärjestelmässä. Skeeman *käytöllä* viitataan tässä yhteydessä esimerkiksi siihen, missä järjestyksessä jonkin luokan ominaisuudet esitetään annotoijalle tai minkälaisia otsikoita (`rdfs:label`) annotaatioille ja muille skeemasta luoduille ilmentymille annetaan. Annotaatioskeeman käyttötapoja kuvaillaan Sahassa *metaskeemassa*, jonka toimintaperiaate muistuttaa osittain Handschuhin ja Staabin (2002) määrittelemää *metaontologiaa*. Erottamalla annotaatioskeeman käyttötapojen kuvailu skeeman varsinaisesta määrittelystä, voidaan jokaiseen skeeman käyttötapahtumaan liittyvät asetukset määritellä erikseen ja säilyttää skeema näin mahdollisimman yleiskäyttöisenä. Metaskeeman ansiosta samaa annotaatioskeemaa voidaan toisin sanoen käyttää useassa eri annotaatioprojektissa ja määritellä tarvittaessa jokaiselle projektille omat skeeman käyttötavat.

Sahan metaskeemassa määritellään projektikohtaisesti annotointiprosessin eri vaiheiden toiminnallisuuteen liittyvät tekijät. Metaskeema kuvaillaan OWL-ontologiana, koska sen avulla pystytään helposti viittaamaan annotaatioskeemassa oleviin resursseihin ja määrittelemään niihin liittyvät asetukset. Seuraavissa kappaleissa on tarkemmin kuvattu metaskeeman toimintaa ja annettu esimerkkejä sen sisältämistä määrityksistä. Liitteessä 2 on esimerkki täydellisestä metaskeemasta.

5.5.2 Aloitusluokat

Sahan luokkasivulla (kuva 15) esitetään hierarkkisesti järjestettynä ne annotaatioskeeman luokat, joista voidaan muodostaa uusia annotaatioita. Tyypillisesti tällaisia luokkia ovat annotaatioskeeman annotaatioluokat (ks. luku 3.3.3). Luokkasivulla näytettävät luokat määritellään annotaatioprojektikohtaisesti projektin metaskeemassa. Metaskeema sisältää luokan `<saha:StartClasses>`, jonka ominaisuus `<saha:annotationClass>` määrittelee annotaatioskeeman luokan, joka näytetään Sahan aloitussivulla. Kun jokin annotaatioskeeman luokka on määritelty annotaatioluokaksi, ovat myös kaikki sen aliluokat annotaatioluokkia. Seuraavassa esimerkissä on metaskeeman osa, jossa määritellään annotaatioluokiksi luokat `atype:SisaltoObjekti` ja `atype:SisaltoProsessi`:

```
<saha:StartClasses rdf:ID="start_classes">
  <saha:annotationClass rdf:resource="atype:SisaltoObjekti"/>
  <saha:annotationClass rdf:resource="atype:SisaltoProsessi"/>
</saha:StartClasses>
```

5.5.3 Otsikko-ominaisuudet

Jotta annotoijan olisi helpompi tulkita annotaatioiden sisältämää ontologisesti kuvattua tietoa, näytetään Sahan käyttöliittymässä resurssin URI-tunnisteen sijaan aina ihmiselle tarkoitettu `rdfs:label`-otsikko, mikäli sellainen on määritelty. Tällaisia otsikoita voidaan määritellä annotaatioskeeman luokille ja ominaisuuksille, sekä luokista luotaville ilmentymille. Kun Sahassa luodaan uusi ilmentymä jostain luokasta, voidaan ilmentymälle muodostaa automaattisesti otsikko sille määriteltyjen ominaisuuksien arvoja hyödyntäen. Tällaista automaattista otsikon muodostamista ohjataan metaskeemalla, jossa määritellään luokkakohtaisesti ne luokan ominaisuudet, joiden arvoja käytetään luokasta luodun ilmentymän `rdfs:label`-kentän arvona.

Ilmentymien otsikoiden määrittely tapahtuu annotaatioprojektin metaskeemassa luokan `saha:ClassDescription` ominaisuuden `saha:labelProperty` avulla. Seuraavassa esimerkissä on määritelty, että luokasta `foaf:Person` (henkilö) luotavien ilmentymien `rdfs:label`-kentän arvoksi tulee ominaisuuksien `foaf:givenname` (etunimi) ja `foaf:surname` (sukunimi) arvot:

```
<saha:ClassDescription rdf:ID="ClassDescription_03">
  <saha:describes rdf:resource="foaf:Person"/>
  <saha:labelProperty rdf:nodeID="labels_01"/>
</saha:ClassDescription>
<rdf:Description rdf:nodeID="labels_01">
  <rdf:type
     rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq"/>
  <rdf:li rdf:resource="foaf:givenname"/>
  <rdf:li rdf:resource="foaf:surname"/>
</rdf:Description>
```

Ominaisuuksien järjestys on määritelty metaskeemassa RDF:n `Seq`-elementtiä käyttäen. Tämän ansiosta ilmentymän otsikko voidaan muodostaa eri ominaisuuksista halutussa järjestyksessä, edellisessä tapauksessa siten, että ensin tulee etunimi ja sitten sukunimi.

Ilmentymän otsikon automaattinen muodostaminen helpottaa annotoijan työtä, koska sen ansiosta uutta ilmentymää luodessaan annotoijan ei erikseen tarvitse määritellä annotointilomakkeella otsikkoa ilmentymälle. Ominaisuus on erityisen hyödyllinen silloin, kun luokalla on literaaliominaisuuksia, jotka kuvailevat sitä ihmiselle mielekkäällä tavalla. Tästä hyvänä esimerkkinä on edellä kuvattu Henkilö-luokka (`foaf:Person`), jolla on ominaisuudet ”etunimi” ja ”sukunimi”.

5.5.4 Näytettävien ominaisuuksien rajoittaminen ja järjestäminen

Sahan annotaatiosivun lomake muodostetaan annotaatioon liittyvän luokan ominaisuuksien perusteella. Jotta ominaisuudet saataisiin lomakkeella haluttuun ja annotoinnin kannalta

loogiseen järjestykseen, määritellään metaskeemassa luokkakohtaisesti luokan ominaisuuksien esitysjärjestys. Metaskeemassa määriteltävällä järjestyksellä ilmaistaan myös ne tietyn luokan ominaisuudet, jotka näytetään annotointilomakkeella. Näin lomakkeesta voidaan haluttaessa jättää pois sellaisia skeemassa määriteltyjä ominaisuuksia, joita ei jostain syystä haluta antaa annotoijan määriteltäväksi.

Seuraavassa esimerkissä on ilmaistu, että luokan `foaf:Person` ominaisuuksista näytetään annotointilomakkeella ensin ominaisuus `foaf:name` ja sitten ominaisuus `foaf:mbox`. Annotaatiolomakkeella näytettäviin ominaisuuksiin viitataan ominaisuudella `saha:annotationProperty`:

```
<saha:ClassDescription rdf:ID="ClassDescription_01">
  <saha:describes rdf:resource="foaf:Person"/>
  <saha:annotationProperty rdf:nodeID="a_01"/>
</saha:ClassDescription>
<rdf:Description rdf:nodeID="a_01">
  <rdf:type
     rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq"/>
  <rdf:li rdf:resource="foaf:name"/>
  <rdf:li rdf:resource="foaf:mbox"/>
</rdf:Description>
```

5.5.5 Julkiset luokat

Saha tukee annotaatioskeeman luokista muodostettujen ilmentymien jakamista eri annotaatioprojektien välillä ONKI-palvelun kautta. Tällaisia luokkia kutsutaan Sahassa julkisiksi luokiksi ja ne määritellään annotaatioprojektin metaskeemassa. Seuraavassa esimerkissä määritellään, että luokka `foaf:Person` on julkinen ja siitä muodostetut ilmentymät lähetetään ONKI-palvelimelle, jonka osoite on `http://demo.seco.tkk.fi/onki/yso/`:

```
<saha:ClassDescription rdf:ID="ClassDescription_01">
  <saha:describes rdf:resource="foaf:Person"/>
  <saha:onkiURL rdf:resource="http://demo.seco.tkk.fi/onki/yso/"/>
</saha:ClassDescription>
```

Ilmentymän lähetys ONKI-palvelimelle tapahtuu silloin, kun uusi ilmentymä luodaan. ONKI:in lähetettyjä ilmentymiä voidaan etsiä ja hakea annotaatioiden arvoiksi vastaavalla tavalla, kuin muita ONKI:ssa olevia referenssiontologioiden käsitteitä.

5.5.6 Objektiominaisuuksien asetusten määrittely

Objektiominaisuuden arvoksi voidaan määritellä Sahassa jostain annotaatioskeeman luokasta muodostettu ilmentymä tai vaihtoehtoisesti jokin ulkopuolisessa ontologiassa määritelty käsite. Mikäli arvoksi määritellään jälkimmäinen, tulee metaskeemassa

määritellä ominaisuuskohtaisesti sen ONKI-palvelimen osoite, josta käsite haetaan. Määrittely tehdään metaskeeman luokan `saha:PropertyDescription` ominaisuudella `saha:onkiURL`. Seuraavassa esimerkissä on ilmaistu, että ominaisuuden `dc:description` arvo haetaan ONKI-palvelimelta, joka sijaitsee osoitteessa `http://demo.seco.tkk.fi/onki/yso/`:

```
<saha:PropertyDescription rdf:ID="d_01">
  <saha:describes rdf:resource="dc:description"/>
  <saha:onkiURL rdf:resource="http://demo.seco.tkk.fi/onki/yso/"/>
</saha:PropertyDescription>
```

Mikäli objektiominaisuudelle ei ole määritelty arvoaluetta (rdfs:range), voidaan sen arvoksi määritellä joko ONKI:n kautta haettava käsite, tai annotaatioskeemassa määritellyn luokan ilmentymä. Metaskeemassa voidaan ilmaista kumpaa näistä halutaan käyttää, tai vaihtoehtoisesti sallia kumpi tahansa. Oletusarvoisesti objektiominaisuudelle, jolle ei ole määritelty arvoaluetta, voidaan valita arvoksi minkä tahansa annotaatioskeeman luokan ilmentymä. Mikäli halutaan sallia myös ONKI:sta haettavat arvot, tulee ONKI:n osoite määritellä edellä esitetyllä tavalla. Jos ominaisuudelle halutaan hakea arvoja ainoastaan ONKI:n kautta, tulee ONKI:n osoitteen lisäksi ilmaista `<saha:allowInstanceValues>`-ominaisuuden avulla (korostettu punaisella), että skeeman luokista muodostettujen ilmentymien käyttö arvona ei ole sallittua:

```
<saha:PropertyDescription rdf:ID="d_01">
  <saha:describes rdf:resource="dc:description"/>
  <saha:onkiURL rdf:resource="http://demo.seco.tkk.fi/stp/onki/yso/"/>
  <saha:allowInstanceValues>false</allowInstanceValues>
</saha:PropertyDescription>
```

5.6 Asetustiedostot

5.6.1 Yleiset asetukset

Jokaiseen annotaatioprojektiin liittyy tiedosto, jossa määritellään projektiin liittyviä asetuksia. Taulukossa 4 on listattu tällaisessa asetustiedostossa määriteltävät asiat. Taulukossa olevat viisi ensimmäistä parametria määrittelevät asetukset tietokannalle, johon annotaatioprojekti on tallennettu. Näiden jälkeen tulevilla parametreilla määritellään onko uusien aliluokkien luonti sallittu sekä millä tavalla annotaatiot yhdistetään annotoitaviin dokumentteihin. Asetustiedosto on tekstitiedosto, jossa jokainen asetus määritellään avain-arvo-parilla. Esimerkki asetustiedostosta on liitteessä 3.

Taulukko 4. *Asetustiedoston parametrit ja niiden tarkoitus*

Parametrin nimi	Tarkoitus
db_user	Tietokannan käyttäjätunnus
db_password	Tietokannan salasana
db_host	Tietokannan osoite
db_port	Tietokannan porttinumero
db_name	Tietokannan nimi
allow_new_subclass	Määrittelee, sallitaanko uusien aliluokkien luonti annotaatioskeeman luokille (arvo true/false)
annotates_property	Määrittelee annotaatioskeeman ominaisuuden, joka kytkee luodut annotaatiot annotoitavaan dokumenttiin. Parametri on valinnainen, sitä ei määritellä dokumentteja luokittelevassa annotoinnissa.

5.6.2 Kieliasetukset

Sahassa tuetaan käyttöliittymän ja ontologioiden monikielisyyttä tarjoamalla annotoijalle mahdollisuus valita sovelluksessa käytettävä kieli. Valittu kieli vaikuttaa Sahassa sekä käyttöliittymäelementtien, että ontologioista (annotaatioskeemasta) haettavien otsikoiden kieleen. Käyttöliittymään liittyvät otsikot määritellään Sahassa kielikohtaisissa asetustiedostoissa, joita on tällä hetkellä laadittu suomen ja englannin kielelle. Myös muita kieliä voidaan ottaa tarvittaessa käyttöön laatimalla niille omat kielitiedostonsa.

Sahassa käytettävään annotaatioskeemaan voidaan määritellä luokkien ja ominaisuuksien otsikoille (rdfs:label) eri kieliversiot käyttämällä XML:n lang-attribuuttia. Kun Saha esittää jonkin skeemassa määritellyn resurssin otsikon käyttöliittymässään, se valitsee resurssille siihen kieleen liittyvän otsikon, jonka Sahan käyttäjä on valinnut aloittaessaan annotoinnin (ks. luku 5.4.2). Mikäli valittua kieltä vastaavaa otsikkoa ei ole määritelty, käytetään joko sellaista otsikkoa, jolle ei ole määritelty kieltä tai vaihtoehtoisesti jollekin muulle kielelle määriteltyä otsikkoa.

5.7 Parametreilla ohjattu käyttö

Annotaatioita voidaan muodostaa Sahassa siten, että annotaatioprojekti, annotaatioluokka ja annotoitava dokumentti määritellään Sahalle lähetettävässä HTTP-pyynnössä, eikä Sahan käyttöliittymässä kuten luvuissa 5.4.2 ja 5.4.3 kuvattiin. Tällainen käyttötapa tulee kyseeseen esimerkiksi silloin, kun Sahaa käytetään jonkin toisen sisällönkuvailu- tai julkaisujärjestelmän rinnalla, jossa annotoitava dokumentti sekä mahdollisesti myös annotaatioluokka on jo määritelty siinä vaiheessa, kun dokumenttia ryhdytään kuvailemaan Sahalla. Parametreilla ohjatussa käytössä annotoija avaa suoraan Sahan annotaatiosivun, jossa ominaisuudet määritellään normaalilla tavalla. Mikäli parametreilla ohjatussa käytössä ei määritellä annotaation luokkaa, valitsee annotoija ensin haluamansa luokan ja sen jälkeen siirtyy annotaatiosivulle.

Annotaatioprojekti, annotoitava dokumentti ja käyttäjän valinnan mukaan myös annotaatioluokka välitetään Sahalle HTTP-protokollan GET-pyynnöllä. Taulukossa 5 on esitetty parametrit, joita käytetään annotaatioprojektin avaamisessa ja uuden annotaation luonnissa. Taulukon alimmaisella rivillä on esimerkki pyynnöstä, jossa nämä parametrit on määritelty.

Taulukko 5. *HTTP-pyynnössä käytettävät parametrit*

Parametrin nimi	**Selitys**	**Esimerkki parametrin arvosta**
model	annotaatioprojektin nimi	test
document	annotoitavan dokumentin URL	http://www.testi.fi
class	annotaation luokka (valinnainen)	http://seco.tkk.fi/projects/saha/annotation/#Test
http://demo.seco.tkk.fi/saha/annotate?model=test&document=http%3A//www.testi.fi&class=http%3A//seco.tkk.fi/projects/saha/annotation/%23Test		

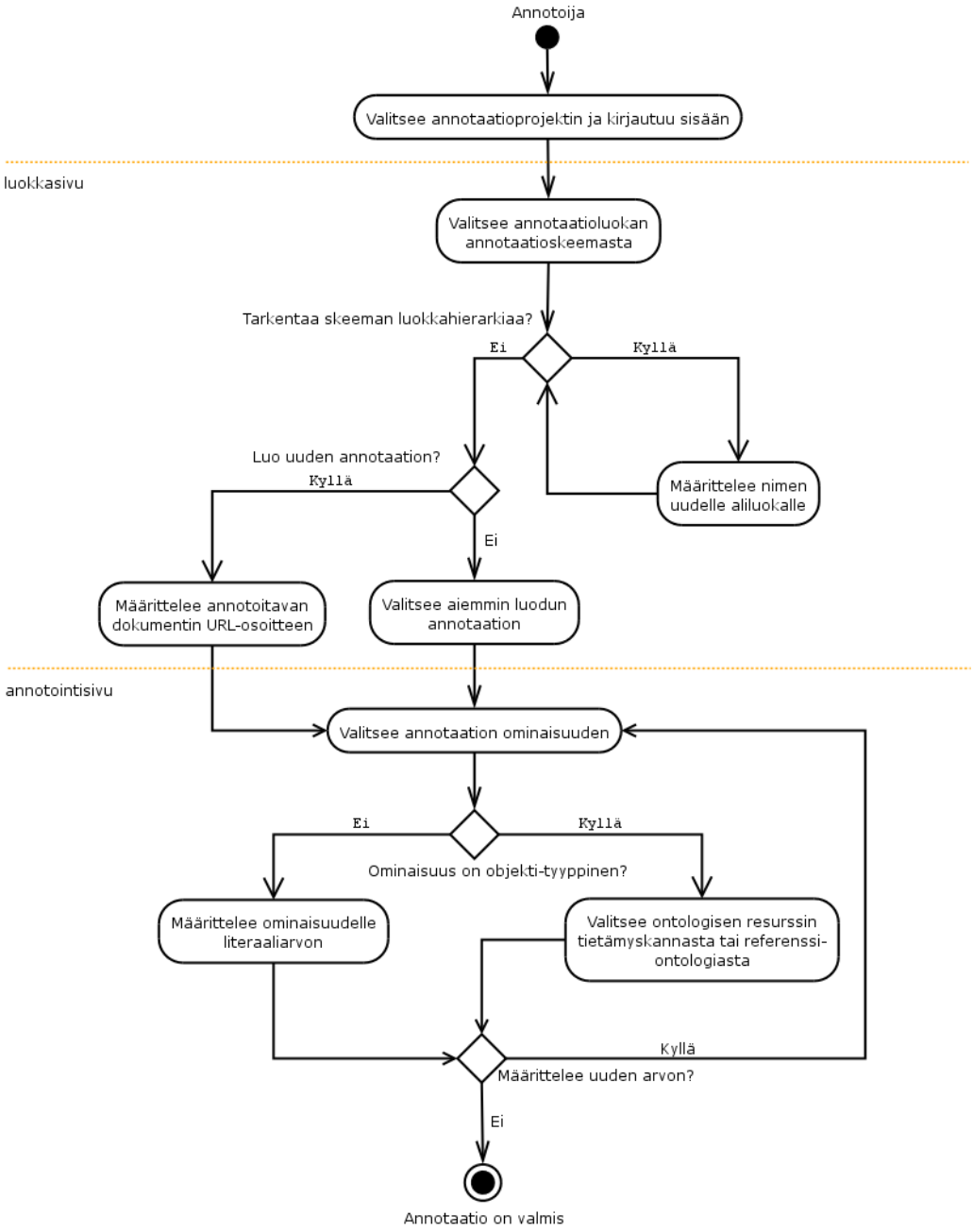
Parametreilla ohjattavaa käyttötapaa voidaan soveltaa esimerkiksi tilanteessa, jossa käytettävää sisällönkuvailujärjestelmää ei pystytä käyttämään sisällön ontologisten kuvailujen tuottamisessa. Koska Saha soveltuu hyvin tällaisen tiedon tuottamiseen, voidaan sillä suorittaa ontologioihin liittyvä osa sisällönkuvailusta. Annotoitavan dokumentin ja annotaatioluokan välittäminen parametreilla Sahalle helpottaa tällaisessa tapauksessa annotoijan työtä, koska niitä ei tarvitse erikseen määrittää uudelleen siirryttäessä käyttämään Sahaa.

5.8 Kokonaiskuvaus annotointiprosessista

Kuvassa 26 on esitetty UML-aktiviteettikaaviolla Sahalla suoritettavan annotointiprosessin kulku. Kaaviossa kuvataan annotointiprosessi yleisellä tasolla, eri vaiheiden tarkemmat kuvaukset on selkeyden vuoksi jätetty pois.

Annotointiprosessi etenee siten, että annotoija valitsee aluksi annotaatioprojektin ja kirjautuu sisään. Tämän jälkeen annotoija siirtyy luokkasivulle, jossa hän näkee annotaatioluokkien hierarkian ja valitsee siitä haluamansa luokan. Mikäli annotoija haluaa tarkentaa skeemassa määriteltyä luokkahierarkiaa, hän luo tässä vaiheessa skeemaan uuden aliluokan valitsemalleen annotaatioluokalle. Luokan valinnan jälkeen annotoija joko valitsee muokattavaksi aikaisemmin luodun annotaation tai luo uuden annotaation määrittelemällä annotoitavan dokumentin osoitteen (URL). Kun annotaatio on valittu tai uusi annotaatio luotu, annotoija siirtyy määrittelemään arvoja annotaation ominaisuuksille. Annotoija valitsee haluamansa ominaisuuden ja määrittelee sille ominaisuuden tyypistä riippuen joko literaali- tai objektiarvon. Objektiominaisuuksien arvot haetaan Sahan tietämyskannasta tai vaihtoehtoisesti jostain ulkoisesta ontologiasta ONKI-palvelun avulla.

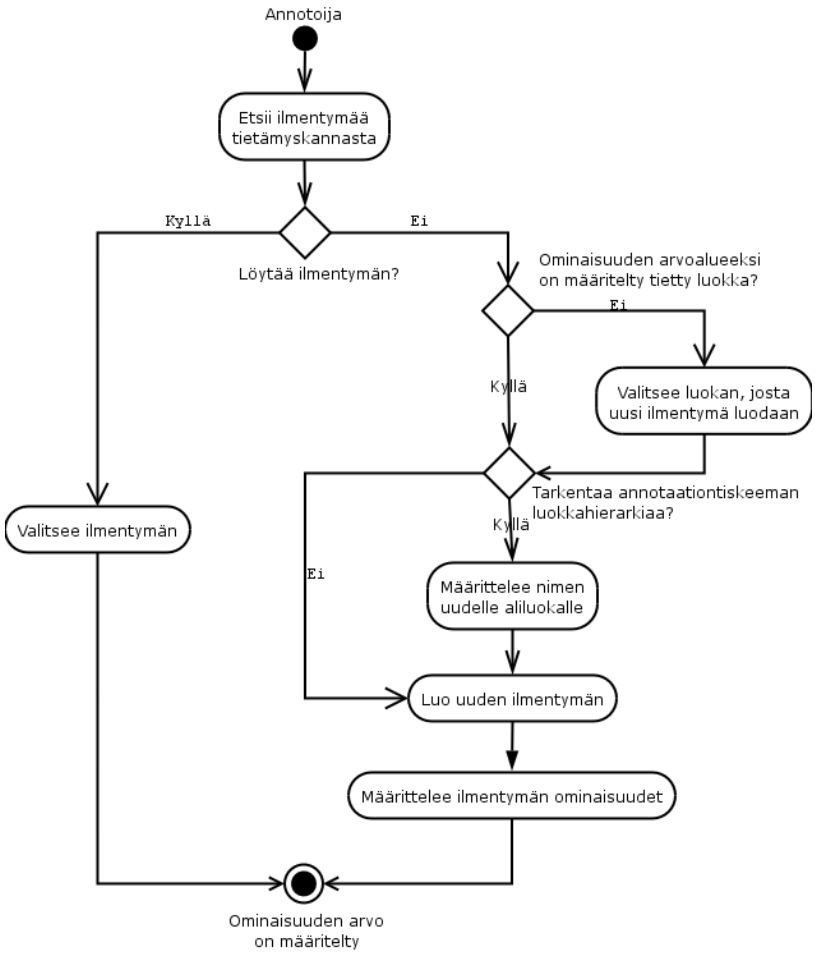
Objektiominaisuuksien määrittäminen on esitetty tarkemmin kuvassa 27. Kun annotoija on määritellyt arvot kaikille haluamilleen ominaisuuksille, annotaatio on valmis.



Kuva 26. *Annotointiprosessin kulku*

Kuvassa 27 on esitetty objektityyppisen ominaisuuden arvon määrittäminen. Tapahtuma on osa annotointiprosessia, joka on esitetty kokonaisuudessaan kuvassa 26. Kuvassa 27 on esitetty tilanne, jossa ominaisuuden arvoksi annetaan jokin annotaatiokeemassa määritellyn luokan ilmentymä. Objektityyppisen ominaisuuden arvoa määrittäessään annotoija etsii aluksi tietämyskannasta sopivaa ilmentymää ja mikäli sellainen löytyy, valitsee sen ominaisuuden arvoksi. Arvoksi valittavien ilmentymien joukkoa voidaan rajoittaa määrittelemällä annotaatioskeemassa ominaisuudelle arvoalue (`rdfs:range`). Mikäli näin on tehty, kohdistuu haku ainoastaan niihin ilmentymiin, joiden tyyppi on sama kuin arvoalueessa määritellyt tyypit. Mikäli arvoaluetta ei ole määritelty, kohdistuu haku

kaikkiin projektissa luotuihin ilmentymiin. Jos haettua ilmentymää ei löydy, annotoija voi luoda uuden ilmentymän, joka asetetaan määriteltävän ominaisuuden arvoksi. Mikäli ominaisuudelle ei ole määritelty arvoaluetta, tai arvoalueeseen kuuluu useampi luokka, valitsee annotoija ensin luokan, josta uusi ilmentymä muodostetaan. Mikäli arvoalueessa on vain yksi luokka, ei tätä valintaa tarvitse tehdä. Seuraavaksi annotoija voi määritellä uuden aliluokan luokalle josta ilmentymää ollaan luomassa, mikäli annotaatioprojektissa on sallittu uusien aliluokkien luonti. Uuden aliluokan luonti on valinnainen toiminto ja se voidaan myös ohittaa. Tämän jälkeen annotoija siirtyy määrittelemään muodostamansa ilmentymän ominaisuuksia. Kun kaikki halutut ominaisuudet on määritelty, on ilmentymän luonti valmis ja alkuperäisen objektiominaisuuden arvo määritetty.



Kuva 27. *Objektityyppisen ominaisuuden arvon määrittäminen*

6 TULOSTEN ARVIOINTIA

Tässä luvussa arvioidaan kehitetyn Saha-annotaatiojärjestelmän toimivuutta, sekä esitetään siihen liittyviä jatkokehitysehdotuksia. Sahaa arvioidaan yleisellä annotointijärjestelmien arviointiin kehitetyllä menetelmällä sekä Sahasta kerättyjen käyttäjäkokemusten perusteella.

6.1 Kehitetty annotointimenetelmä

Työssä kehitettiin annotaatioskeemoja sekä ontologioita hyödyntävä annotointimenetelmä, jonka avulla voidaan kuvailla erilaisia webissä olevia dokumentteja sekä niihin liittyviä resursseja. Skeemojen hyödyntämistä annotoinnissa voidaan pitää perusteltuna mm. siksi, että niiden avulla pystytään vapaata annotointia selkeämmin ilmaisemaan annotaation suhde kuvailtavaan resurssiin. Skeemat havaittiin lisäksi toimivaksi tavaksi ohjata ja ennen kaikkea helpottaa annotoijan näkökulmasta annotaatioiden muodostamista.

Skeemojen avulla annotointia todettiin olevan mahdollista suorittaa siten, että annotoijan ei tarvitse tulkita ontologioihin liittyviä monimutkaisia luokkahierarkioita ja niiden perusteella tehdä päätöstä siitä, minkälaisia asioita resursseista kuvaillaan. Tätä pidettiin tarpeellisena erityisesti sellaisten annotoijien kohdalla, jotka eivät tunne tarkemmin semanttisen webin tekniikoita. Skeemojen tärkeänä ominaisuutena nähtiin myös se, että niiden avulla voidaan tehokkaasti rajoittaa ja täsmentää sitä, minkälaisia arvoja annotaatioissa tulee käyttää. Joissain tilanteissa skeemojen rajoittavuus saattaa tosin muodostua ongelmaksi, mikäli rajoitusten takia niillä ei pystytäkään ilmaisemaan annotoitavasta aineistosta kaikkia haluttuja asioita. Skeemaperustaisessa annotoinnissa saatetaan kohdata myös yhteensopivuusongelmia, jotka muodostuvat eri annotaatioskeemojen pohjalta tuotettujen annotaatioiden välille.

6.2 Saha-järjestelmä

6.2.1 Ominaisuudet

Sazedj ja Pinto (2005) kuvailevat yleisen arviointimenetelmän, jonka avulla voidaan vertailla erilaisten annotaatiotyökalujen ominaisuuksia ja käytettävyyttä. Menetelmä perustuu 20 arviointikriteeriin, jotka on jaettu sovellusalueesta riippumattomiin (domain-independent) ja riippuviin (domain-specific) kriteereihin. Ensimmäisiin kuuluvat sovelluksen käyttöliittymään (mm. käytettävyys, yksinkertaisuus) ja yleisiin ominaisuuksiin (mm. dokumentaatio, skaalautuvuus, vakaus) liittyvät arviointikriteerit. Jälkimmäisiin taas kuuluvat sovelluksella tuotettavaan metadataan sekä sovelluksen toimintaan liittyvät kriteerit. Jokainen arviointikriteeri sisältää joukon ominaisuuksia, joiden perusteella arvioitavia järjestelmiä voidaan pisteyttää siten, että järjestelmä saa jokaisesta tukemastaan ominaisuudesta pisteen. Taulukossa 6, joka on muodostettu Sazedj

ja Pinton laatiman taulukon mukaan, on arvioitu Sahan ominaisuuksia sovellusalueesta riippuvilla kriteereillä. Taulukosta on jätetty pois sellaiset arviointikriteerit, joiden käyttäminen ei ole mielekästä Sahan arvioinnissa. Näitä ovat mm. automatisointiin liittyvät kriteerit (tarkkuus, saanti, luotettavuus ja nopeus). Taulukon oikeassa reunassa olevassa sarakkeessa kerrotaan kunkin kriteerin kohdalla, mitä ominaisuuksia Sahassa tuetaan sekä mahdollisesti perustellaan, miksi jotain ominaisuutta ei ole tuettu.

Taulukko 6. *Sahan ominaisuuksien arviointia*

Arviointi-kriteeri	**Määritelmä**	**Ominaisuudet**	**Saha**
Annotaatioiden liittäminen (*association*)	Tapa, jolla annotaatio liitetään annotoitavaan resurssiin	(1) Työkalu tukee dokumentista irrallisia annotaatioita. (2) Työkalu tukee dokumenttiin liitettyjä annotaatioita.	Sahassa toteutuu kohta (1). Annotaatioiden hajautetun tuottamisen ja keskitetyn hallinnan vuoksi annotaatioita ei tallenneta osaksi dokumenttia.
Joustavuus (*flexibility*)	Kyky olla mukautuva tai muuntautuva	(1) Aiemmin luotuja annotaatioita on mahdollista poistaa. (2) Ontologioiden muokkaus on mahdollista. (3) Annotaatioita varten voidaan määrittää oma nimiavaruus	Saha toteuttaa kohdat (1) ja (3). Ontologioiden muokkaaminen (pl. aliluokkien luonti) jätettiin pois, koska käyttöliittymän toiminnot haluttiin pitää mahdollisimman yksinkertaisina.
Eheys (*integrity*)	Kyky varmistaa annotaatioiden mielekkyys ja yhtenäisyys	(1) Kuvailtavan suhteen arvoalue (range) varmistetaan. (2) Annotaation oikeellisuus tarkastetaan ajan kuluessa. (3) Annotaation ja resurssin assosiaatio tarkistetaan.	Saha toteuttaa annotaatioskeeman rakenteesta riippuen kohdan (1). Assosiaatioita ei tarkisteta, koska Sahalla voidaan annotoida myös sellaisia resursseja, joita ei voida hakea esim. URL:n avulla. Annotaation oikeellisuuden tarkistaminen liittyy automaattiseen annotointiin.
Kohde (*scope*)	Tarkkuus, jolla annotaatio voidaan kohdistaa tiettyyn osaan annotoitavassa resurssissa	(1) Annotoidaan pienin mahdollinen informaatioyksikkö. (2) Annotoidaan mikä tahansa edellisen moninkerta. (3) Annotoidaan resurssi kokonaisuutena.	Saha toteuttaa kohdan (3).
Annotoitavat resurssit (*input*)	Minkälaisten resurssien annotointia tuetaan	(1) Web-sivu (2) Tekstidokumentti (3) Kuva	Saha toteuttaa kaikki kolme kohtaa. Koska annotaatio on täysin irrallinen dokumentista, riittää että dokumenttiin voidaan viitata URI-tunnisteella.
Annotaatiot (*output*)	Kyky tuottaa annotaatioita formaatissa, joka on käyttökelpoinen semanttisessa webissä.	(1) Annotaatiot on kuvattu W3C:n standardien mukaan. (2) Annotaatiot on yksikäsitteisesti liitetty ontologisiin käsitteisiin.	Saha toteuttaa kummatkin kohdat.

Sazedj'n ja Pinton arviointimenetelmää ja sillä saatuja tuloksia tarkasteltaessa on huomioitava, että kaikki menetelmän arviointikriteerit eivät ole riittävän yksikäsitteisiä ja

yleisiä, jotta ne soveltuisivat minkä tahansa annotaatiosovelluksen arviointiin. Menetelmällä tehty arvio kuvaa kuitenkin suuntaa-antavasti sitä, miten hyvin arvioitu sovellus tukee niitä ominaisuuksia, joita annotaatiosovelluksesta tyypillisesti odotetaan löytyvän. Tarkasteltaessa taulukossa 6 olevia tuloksia, havaitaan Sahan tukevan pääpiirteittäin hyvin eri arviointikriteereihin liittyviä ominaisuuksia.

6.2.2 Käytettävyys

Saha on ollut koekäytössä FinnONTO-projektissa[30] kehitettävän semanttisen terveystietoportaali Tervesuomi.fi:n[31] (Holi ym. 2006) metadatan tuottamisessa. Siinä Terveyden edistämisen keskus ry:lle (Tekry) laadittiin annotaatioskeema, jonka pohjalta keskuksen informaatikko annotoi joukon Tekryn toimintaan liittyviä verkkosivuja. Annotoinnin jälkeen informaatikko vastasi kyselyyn (liite 4), jossa esitettiin ohjelman toimintaan, käytettävyyteen sekä annotointiin yleisesti liittyviä kysymyksiä. Kyselyn vastausten perusteella kävi ilmi, että Sahan käyttö koettiin yleisesti selkeäksi, eikä annotoinnissa kohdattu muutamia teknisiä ongelmia lukuun ottamatta suurempia vaikeuksia. Eniten epäselvyyksiä aiheutti käytössä ollut annotaatioskeema, josta annotoija ei kunnolla osannut valita annotoitavaa dokumenttia kuvaavaa luokkaa. Annotoijan mielestä luokat oli nimetty siten, että niiden välisiä eroja ei aina ollut helppo hahmottaa. Skeeman annotaatioluokilla oli suhteellisen paljon ominaisuuksia (19 ominaisuutta/luokka), minkä johdosta annotointisivu kasvoi annotoijan mielestä turhan suureksi. Edellä kuvattujen ongelmien pohjalta voidaan todeta, että skeeman suunnittelulla pystytään merkittävästi vaikuttamaan annotoinnin sujuvuuteen. Annotaatioluokkien huolellinen nimeäminen on tärkeää ja niiden valintaa voidaan helpottaa myös rajoittamalla luokkien määrää. Lisäksi olisi hyödyllistä, että annotoijalle annettaisiin ennen annotoinnin alkamista hyvä kirjallinen tai suullinen kuvaus annotaatioskeeman rakenteesta ja ohjeet siitä, miten sitä tulee käyttää.

Sahan käytettävyyttä ei ole tutkittu vielä riittävästi, jotta siitä pystyttäisiin antamaan tarkempia tuloksia. Palautteen kerääminen Sahan käyttäjiltä sekä mahdollisesti myös käyttäjätestit ovat tarpeellisia, mikäli sovelluksen käytettävyyttä ryhdytään parantamaan.

6.2.3 Teknisten ratkaisujen toimivuus

Sahassa käytetyt tekniset ratkaisut osoittautuivat toimiviksi ja prototyyppivaiheen vaatimustasoon nähden riittäviksi. Apachen Cocoon-ympäristö antoi mahdollisuuden mallintaa sovelluksen toimintaa web-sovelluksen sijasta perinteisemmän sovelluksen tapaan ja se helpotti monien toimintojen toteuttamista. Cocoonin havaittiin toisaalta sisältävän paljon ominaisuuksia, joita Sahassa ei hyödynnetä ja siksi jokin kevyempi palvelinohjelmisto voisi olla Sahan yhteydessä tehokkaampi ja ennen kaikkea yksinkertaisempi ratkaisu.

---

[30] http://www.seco.tkk.fi/projects/finnonto/

[31] http://www.seco.tkk.fi/applications/tervesuomi/

Jena-ympäristön käyttö sekä annotaatioiden tallentaminen tietokantaan havaittiin toimiviksi ratkaisuiksi. Kun ohjelmaa kokeiltiin erilaisilla annotaatioskeemoilla, todettiin monimutkaisempien skeemojen hidastavan jonkin verran ohjelman toimintaa. Vaikka hidastumisella ei ollut suoritetuissa kokeissa merkittävää vaikutusta ohjelman käytettävyyteen, on asiaa perusteltua tutkia tarkemmin ohjelman jatkokehityksessä.

Sahan käyttöliittymän toteutuksessa Ajax-tekniikat olivat olennaisessa osassa. Niiden avulla käyttöliittymään voitiin toteuttaa ominaisuuksia, jotka eivät ole mahdollisia perinteisillä www-tekniikoilla. Ajaxin avulla toteutettiin mm. erilaiset instanssi- ja käsitehaut (semantic autocompletion), jotka olennaisesti helpottavat ja selkeyttävät annotoinnin perustana olevien ontologioiden hyödyntämistä. Lisäksi Ajax-tekniikoiden avulla ohjelman käytettävyyttä ja nopeutta pystyttiin merkittävästi parantamaan. Useimmat käyttäjän valinnoista riippuvat toiminnot suoritetaan Ajaxin avulla taustalla, eikä käyttäjä siten havaitse niiden suorittamiseen liittyvää viivettä. Ajaxin avulla toteutetut toiminnot osoittavat, että web-sovellukset voidaan nykyään toteuttaa käytettävyydeltään ja toiminnoiltaan yhä enemmän perinteisiä sovelluksia vastaavasti.

6.2.4 Jatkokehitysehdotukset

Sahan jatkokehitysehdotukset voidaan jakaa kolmeen eri ryhmään, joita ovat toiminnallisuuden, teknisten ratkaisujen sekä käytettävyyden kehittäminen.

Toiminnallisuuteen liittyvistä kehityskohteista yhdeksi tärkeimmistä nousee tietoturva. Sahan prototyyppivaiheen toteutuksessa ei ole mukana kaikkia turvallisuuteen liittyviä piirteitä, kuten esimerkiksi mahdollisuutta hallita annotaatioprojekteihin pääsyä ja niissä olevien annotaatioiden hakua käyttäjäkohtaisilla salasanoilla. Toinen merkittävä kehityskohde on automaation soveltaminen annotoinnissa. Tässä yhteydessä varteenotettava vaihtoehto on Poka-järjestelmä (ks. luku 5.1), joka sisältää monipuolisia työkaluja luonnollisen kielen tekstin käsittelyyn ja erilaisten semanttisten rakenteiden automaattiseen tunnistamiseen. Pokan hyödyntämisen lisäksi Sahaan on mahdollista toteuttaa myös kevyempiä automatisointiin liittyviä ratkaisuja, kuten esimerkiksi HTML-dokumenttien otsikko- ja metatietoelementtien tunnistamista ja niiden automaattista sisällyttämistä annotaatioihin. Toiminnallisuuteen liittyviin kehityskohteisiin kuuluvat myös Sahan hallintatoiminnot, joiden avulla ohjataan uusien annotaatioprojektien luontia ja projekteissa muodostettujen annotaatioiden välittämistä eteenpäin niitä hyödyntäville sovelluksille. Uusien annotaatioprojektien luontia voidaan helpottaa kehittämällä hallintakäyttöliittymän toimintoja esimerkiksi siten, että käyttäjä voi määrittää projektin asetukset asetustiedoston sijaan www-lomakkeella ja lähettää lomakkeen avulla myös annotaatioskeeman palvelimelle. Jatkokehitystä tarvittaisiin lisäksi HTTP-GET-rajapinnassa, jonka avulla eri sovellukset voivat noutaa Sahassa olevia annotaatioita. Tällä hetkellä rajapinnan kautta on mahdollista hakea ainoastaan kokonainen annotaatioprojekti, mutta ei yksittäistä annotaatiota tai ilmentymää, mikä saattaisi olla hyödyllistä joidenkin sovellusten kohdalla.

Teknisiin kehityskohteisiin kuuluvat mm. ohjelman suorituksessa tapahtuvien poikkeus- ja

virhetilanteiden hallinnan parantaminen. Tällä hetkellä kaikista virhetilanteista ei näytetä annotoijalle selkeää ilmoitusta, eikä välttämättä tarjota johdonmukaista tapaa jatkaa ohjelman käyttöä virheen jälkeen. Virhetilanteista kertovien tietojen tallentamista ja raportointia esimerkiksi lokitiedostojen avulla sovelluksen ylläpidolle tulisi niin ikään kehittää. Ohjelman skaalautuvuutta suurempiin käyttäjä- ja tietomääriin tulisi tutkia ja tehdä tarvittaessa siihen liittyviä teknisiä parannuksia.

Sahan käytettävyydestä ja siihen liittyvistä ongelmista tulisi kerätä tarkempia tietoja ja niiden pohjalta kehittää sovelluksen käyttöliittymää. Tällä hetkellä käyttöliittymän ulkoasu on laadittu prototyyppivaihetta varten, eikä sen suunnittelussa ole näin ollen otettu huomioon kaikkia hyvään käytettävyyteen vaikuttavia tekijöitä. Palautetta ohjelman käytettävyydestä tulisi kerätä ensisijaisesti ohjelmaa käyttäviltä annotoijilta ja tarpeen vaatiessa toteuttaa myös tarkoitusta varten suunniteltuja käyttäjätestejä.

7 YHTEENVETO

Tässä diplomityössä tutkittiin semanttisen webin ontologiaperustaista annotointia ja kehitettiin menetelmä webissä olevien dokumenttien annotointiin. Työssä selvitettiin aluksi, minkälaisia annotaatioita webin yhteydessä tuotetaan ja jaettiin annotaatiot luonnollista kieltä sisältäviin tekstiannotaatioihin sekä formaalisti kuvattuihin ontologiaperustaisiin annotaatioihin. Tekstiannotaatiot ovat tarkoitettu ihmisten luettaviksi, eikä niillä sen vuoksi ole varsinaista merkitystä semanttisen webin näkökulmasta. Ontologioihin pohjautuvat annotaatiot puolestaan muodostavat pohjan erilaisille semanttisen webin sovelluksille, kuten esimerkiksi portaaleille ja hakupalveluille. Ontologioihin perustuvissa annotaatioissa tieto on kuvattu formaalisti yhteisesti sovittuja käytäntöjä ja standardeja noudattaen.

Annotaatioiden muodostamisessa käytettävät menetelmät jaettiin vapaaseen ja skeemaperustaiseen annotointiin. Vapaassa annotoinnissa resursseja kuvaillaan ontologisilla käsitteillä, joita liitetään annotoitavaan resurssiin. Käytännössä annotointi voi olla esimerkiksi kuvailtavassa tekstissä olevien sanojen merkkaamista ontologioissa määritellyillä käsitteillä. Skeemaperustaisessa annotoinnissa annotaatioiden rakenteen määrittelee annotaatioskeema, joka voidaan kuvata ontologiana. Skeemoja hyödyntävässä annotoinnissa päästään yleensä vapaata annotointia rikkaampaan ilmaisuun, koska skeeman avulla voidaan monipuolisesti kuvailla esimerkiksi erilaisia tekstissä esiintyviä semanttisia suhteita. Annotaatioskeemat auttavat tarvittaessa määrittelemään yksikäsitteisesti asiat, jotka annotoitavista resursseista halutaan kuvailla ja niiden avulla voidaan tehokkaasti varmistaa annotaatioiden yhteentoimivuus.

Työssä esiteltiin aikaisemmin kehitettyjä annotaatiojärjestelmiä ja analysoitiin niiden käyttökelpoisuutta hajautetussa metadatan tuottamisessa, jossa annotoijilla ei ole tuntemusta semanttiseen webiin liittyvistä tekniikoista ja erikoiskäsitteistä. Sovelluksia tutkittaessa havaittiin, että ne vaativat tyypillisesti vähintään perustason tietämystä RDF-perustaisista kielistä sekä ontologioista. Lisäksi havaittiin, että sovellukset eivät pääsääntöisesti tue kovin hyvin annotointiprosessin jakamista useiden eri annotoijien kesken. Koska voidaan olettaa, että semanttisen webin sisältöjä kuvailee tulevaisuudessa suuri joukko myös sellaisia henkilöitä, joilla ei ole asiantuntemusta semanttisen webin tekniikoista ja että tällaista sisällönkuvailutyötä tehdään hajautetusti eri tahoilla, tunnistettiin tarve kehittää annotointijärjestelmä, jonka suunnittelussa ja toteutuksessa nämä seikat on otettu huomioon.

Semanttisen webin annotointia tukevalle sovellukselle määritettiin yleiset perusvaatimukset, joita ovat monimutkaisten ontologioiden sekä niihin liittyvien erikoiskäsitteiden piilottaminen käyttäjältä, annotoinnin hajauttaminen, skeemojen käyttö, sekä selainpohjaisuus. Saha on annotointisovellus, jonka avulla pystytään ontologiapohjaisesti kuvailemaan erilaisia webissä olevia dokumentteja. Saha on toteutettu web-sovelluksena, jota voidaan käyttää tavallisella web-selaimella. Sahan avulla

annotointiprosessi voidaan hajauttaa useille käyttäjille, jotka pystyvät hyödyntämään ja muokkaamaan toistensa tekemiä annotaatioita. Sovelluksen käyttöliittymä on suunniteltu siten, että sen käyttö ei vaadi semanttiseen webiin liittyvien tekniikoiden ja erikoiskäsitteiden tuntemista. Sahalla muodostettavissa annotaatioissa voidaan hyödyntää ulkopuolisissa ontologioissa määriteltyjä käsitteitä kytkemällä se ONKI-ontologiapalveluun. ONKI:n avulla voidaan jakaa myös annotoinnin yhteydessä muodostettuja ilmentymiä annotoijien välillä. Sahaa on testattu FinnONTO-projektiin liittyvien semanttisten portaalien sisällönkuvailussa. Saadut tulokset ovat olleet lupaavia, mutta testausta ja sovelluksen kehittämistä tulisi edelleen jatkaa.

LÄHDELUETTELO

Antoniou, G., van Harmelen, F. (2004) *A Semantic Web Primer*. Cambridge, Massachusetts, The MIT Press. 238s.

Auer, S. (2005) *Powl – A Web Based Platform for Collaborative Semantic Web Development*. Proceedings of the Workshop on Scripting for the Semantic Web, Heraklion, Kreikka, 30.5.2005.

Beckett, D., McBride, B. (2004) *RDF Syntax Specification (Revised)* (online). World Wide Web Consortium (W3C). Päivitetty 10.2.2004 [viitattu 14.4.2006]. Saatavilla www-muodossa: <URL: http://www.w3.org/TR/rdf-syntax-grammar/>.

Berners-Lee, T., Hendler, J., Lassila, O. (2001) The Semantic Web, *Scientific American*, 5/2001.

Booth, D. (2003). *Four Uses of a URL: Name, Concept, Web Location and Document Instance* (online). World Wide Web Consortium (W3C). Päivitetty 28.1.2003 [viitattu 6.3.2006]. Saatavilla www-muodossa: <URL:http://www.w3.org/2002/11/dbooth-names/dbooth-names_clean.htm>.

Brickley, D., Guha, R.V. (2004) *RDF Vocabulary Description Language 1.0: RDF Schema* (online). World Wide Web Consortium (W3C). Päivitetty 10.2.2004 [viitattu 6.3.2006]. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/rdf-schema/>.

Cimiano, P., Ciravegna, F., Dominigue, J., Handschuh, S., Lavelli, A., Staab, S., Stevenson, M. (2003) *Requirements for Information Extraction for Knowledge Management*. Proceedings of the Knowledge Markup and Semantic Annotation Workshop, 2nd International Conference on Knowledge Capture KCAP–2003.

Ciravegna, F., Wilks, Y (2003) Designing Adaptive Information Extraction for the Semantic Web in Amilcare. Teoksessa: Handschuh, S., Staab, S. (toim.). *Annotation for the Semantic Web*. Amsterdam, IOS Press, s. 25–45.

Corcho, O. (2006) *Ontology–based Document Annotation: Trends and Open Research Problems*, International Journal of Metadata, Semantics and Ontologies, 1, 1, 2006.

Dill, S., Tomlin, J., Zien, J., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A. (2003) *SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation.* Proceedings of the 12th International World Wide Web Conference (WWW 2003), Budapest, Unkari, 20.–24.5.2003.

Fensel, D., Hendler, J., Lieberman, H., Wahlster, W. (2003) *Spinning the Semantic Web*. Cambridge, Massachusetts, The MIT Press.

Geurts, J., van Ossenbruggen, J., Hardman, L. (2005) *Requirements for Practical Multimedia Annotation.* Multimedia and the Semantic Web Workshop, 2nd European Semantic Web Conference ESWC2005, Heraklion, Kreikka, 29.5.–1.6.2005.

Grishman, R. (1997) Information Extraction: Techniques and Challenges. Teoksessa: Pazienza, M.T. (toim.). *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*. Heidelberg, Springer-Verlag. s. 10–27.

Gruber, T. (1993) *A Translation Approach to Portable Ontology Specifications.* Knowledge Acquisition 5, 2. s. 199–220.

Haarslev, V., Möller, R. (2003) *Racer: An OWL Reasoning Agent for the Semantic Web*. Proceedings of the International Conference on Web Intelligence, Workshop on Applications, Products and Services of Web-based Support Systems. Halifax, Kanada, 13.10.2003.

Handschuh, S., Staab, S. (2002) *Authoring and Annotation of Web Pages in CREAM*. Proceedings of the 11th International Conference on World Wide Web, Honolulu, USA, 7.–11.5.2002.

Handschuh, S., Staab, S., Ciravegna, F. (2002) *S–CREAM – Semi–automatic Creation of Metadata*. Proceedings of 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), Siguenza, Espanja, 1.–4.10.2002.

Handschuh, S., Staab, S. (2003) Annotation of the Shallow and the Deep Web. Teoksessa: Handschuh, S., Staab, S. (toim.). *Annotation for the Semantic Web*. Amsterdam, IOS Press, s. 25–45.

Hawke, S. (2001) *How We Identify Things (on the Semantic Web)?* (online) World Wide Web Consortium (W3C). Päivitetty 21.9.2001 [viitattu 6.3.2006]. Saatavilla www-muodossa: <URL:http://www.w3.org/2001/03/identification-problem/>.

Heflin, J. (2004) *OWL Web Ontology Language Use Cases and Requirements* (online). World Wide Web Consortium (W3C). Päivitetty 10.2.2004 [viitattu 6.3.2006]. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/webont-req/>.

Holi, M., Lindgren, P., Suominen, O., Viljanen, K., Hyvönen, E. (2006). *TerveSuomi.fi – A Semantic Health Portal for Citizens*. Proceedings of the 1st Asian Semantic Web Conference (ASWC2006). Peking, Kiina, 3.–7.9.2006.

Hyvönen, E., Valo, A., Komulainen, V., Seppälä, K., Kauppinen, T., Ruotsalo, T., Salminen, M., Ylisalmi, A. (2005) *Finnish National Ontologies for the Semantic Web - Towards a Content and Service Infrastructure*. Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005), Madrid, Espanja, 12.–15.9.2005.

Hyvönen, E., Mäkelä, E. (2006) *Semantic Autocompletion*. Proceedings of the First Asian Semantic Web Conference (ASWC 2006), Peking, Kiina, 3.–7.9.2006. Springer-Verlag.

Hyvönen, E., Saarela, S., Viljanen, K. (2004) *Application of Ontology Techniques to View–Based Semantic Search and Browsing*. The Semantic Web: Research and Applications, First European Semantic Web Symposium, ESWS 2004. Heraklion, Kreikka, 10.–12.5.2004. Lecture Notes in Computer Science 3053, Springer. s. 92–106.

Hyvönen, E., Junnila, M., Kettula, S., Saarela, S., Salminen, M., Syreeni, A., Valo, A., Viljanen, K. (2003) *Publishing Collections in the Finnish Museums on the Semantic Web Portal – First Results*. Proceedings of XML Finland 2003, Kuopio, Suomi, 30.–31.10. 2003.

Kahan, J., Koivunen, M.R., Prud'Hommeaux, E., Swick R.R. (2001) *Annotea: An Open RDF Infrastructure for Shared Web Annotations*, Proceedings of the 10th International World Wide Web Conference (WWW10), Hong Kong, Kiina, 1.–5.5.2001.

Kalyanpur, A., Hendler, J., Parsia, B., Golbeck, J. (2005) *SMORE – Semantic Markup, Ontology, and RDF Editor*. (online) University of Maryland Institute for Advanced Computer Studies. [viitattu 6.3.2006] Saatavilla www-muodossa: <URL:http://www.mindswap.org/papers/SMORE.pdf>.

Kettler, B., Starz, J., Miller, W., Haglich, P. (2005) *A Template–based Markup Tool for Semantic Web Content.* 4th International Semantic Web Conference ISWC2005, Galway, Irlanti, 6.–10.10.2005. Lecture Notes in Computer Science 3729, Springer. s. 446–460.

Kiriyakov, A., Popov, B., Ogynyanoff, D., Manov, D., Kirilov, A., Goranov, M. (2003) *Semantic Annotation, Indexing, and Retrieval*. 2nd International Semantic Web Conference ISWC 2003, Florida, USA, 20.–23.10.2003.

Komulainen, V., Valo, A., Hyvönen, E. (2005) *A Tool for Collaborative Ontology Development for the Semantic Web.* Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005), Madrid, Espanja, 12.–15.9.2005.

Manola, F., Miller, E. (2004) *RDF Primer* (online). World Wide Web Consortium (W3C). Päivitetty 10.2.2004 [viitattu 6.3.2006]. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/rdf-primer/>.

Marshall, C. C. (1998) *Toward an Ecology of Hypertext Annotation*, Proceedings of ACM Hypertext'98, 9th Conference on Hypertext and Hypermedia, Pittsburgh, USA, 6/1998.

McQuinness, D., van Harmelen, F. (2004) *OWL Web Ontology Language Overview* (online). World Wide Web Consortium (W3C). Päivitetty 10.2.2004 [viitattu 6.3.2006]. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/owl-features/>.

Naing, M.M., Limy, E-P., Hoe-Lian, D. (2002) *Ontology–based Web Annotation Framework for HyperLink Structures.* Proceedings of 3rd International Conference on Web Information Systems Engineering Workshops (WISE 2002 Workshops), Singapore, 11.12.2002. IEEE Computer Society 2002. s. 184–193.

Noy, N., Sintek, M., Decker, S., Crubézy, M., Fergerson, R. (2001) Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems* 2, 16. s. 60–71.

Reeve, L., Han, H. (2005) *Survey of Semantic Annotation Platforms*. 20th ACM Symposium on Applied Computing, Santa Fe, USA, 13.–17.3.2005.

RFC 3986 (2005) *Uniform Resource Identifier (URI): Generic Syntax* (online). The Internet Engineering Task Force (IETF). Päivitetty 5/2005 [viitattu 6.3.2006]. Saatavilla www-muodossa: <URL:http://www.ietf.org/rfc/rfc3986.txt>.

Sazedj, P., Pinto, H.S. (2005) *Time to evaluate: Targeting Annotation Tools*. 4th International Semantic Web Conference ISWC2005, Knowledge Markup and Semantic Annotation Workshop, Galway, Irlanti, 6.–10.10.2005.

Schreiber, G., Dubbeldam, B., Wielemaker, J., Wielinga, B. (2001) Ontology–Based Photo Annotation. IEEE Intelligent Systems, 16, 3, s. 66–74.

Silva, N., Rocha, J. (2003) *Semantic Web Complex Ontology Mapping*. Proceedings of IEEE/WIC International Conference on Web Intelligence. Halifax, Kanada, 13.–16.10.2003

Studer, R., Benjamins, V.R., Fensel, D. (1998) Knowledge Engineering: Principles and Methods. IEEE Transactions on Data and Knowledge Engineering, 25, 1–2, s. 161–197.

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F. (2006) Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. *Journal of Web Semantics*, 4, 1, s. 14–28.

Valkeapää, O., Hyvönen, E. (2006) *A Browser–based Semantic Annotation Tool for Distributed Content Creation*. Proceedings of the 1st Asian Semantic Web Conference (ASWC2006), Workshop on Semantic Web Applications and Tools. Peking, Kiina, 3.–7.9.2006.

Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., Ciravegna, F. (2002) *MnM: Ontology Driven Semi–Automatic and Automatic Support for Semantic Markup.* 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), Siguenza, Espanja, 1.–4.10.2002.

LIITE 1. Esimerkki annotaatioskeemasta

Alla on esimerkki yksinkertaisesta annotaatioskeemasta, jossa kuvaillaan erilaisia dokumenttityyppejä ja niihin liittyviä ominaisuuksia.

```
<?xml version="1.0"?>
<rdf:RDF

<!-- Nimiavaruudet: -->
    xmlns="http://www.seco.hut.fi/def-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:schema="http://www.seco.hut.fi/annotation#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://www.seco.hut.fi/def-ns">

  <owl:Ontology rdf:about=""/>

<!-- Luokat: -->
  <owl:Class rdf:about="foaf:Person"/>
  <owl:Class rdf:about="schema:Article">
    <rdfs:subClassOf>
      <owl:Class rdf:about="schema:DocumentType"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="schema:Book">
    <rdfs:subClassOf rdf:resource="schema:DocumentType"/>
  </owl:Class>
  <owl:Class rdf:about="schema:WebPage">
    <rdfs:subClassOf rdf:resource="schema:DocumentType"/>
  </owl:Class>

<!-- Ominaisuudet: -->
  <owl:ObjectProperty rdf:about="dc:creator">
    <rdfs:range rdf:resource="foaf:Person"/>
    <rdfs:domain rdf:resource="schema:DocumentType"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:about="foaf:name">
    <rdfs:domain rdf:resource="foaf:Person"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="dc:description">
    <rdfs:domain rdf:resource="schema:DocumentType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="foaf:mbox">
    <rdfs:domain rdf:resource="foaf:Person"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="dc:title">
    <rdfs:domain rdf:resource="schema:DocumentType"/>
  </owl:DatatypeProperty>
</rdf:RDF>
```

LIITE 2. Esimerkki Sahan metaskeemasta

Alla on esimerkki Sahan metaskeemasta. Metaskeema liittyy aina johonkin tiettyyn annotaatioskeemaan ja sitä kautta annotaatioprojektiin. Metaskeemassa kuvaillaan annotaatioskeeman käyttötavat projektissa.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:saha="http://www.seco.tkk.fi/projects/finnonto/saha/saha#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">

<owl:Ontology rdf:about=""/>

<!-- metaskeeman luokkien ja ominaisuuksien kuvaus: -->

<owl:Class rdf:about="saha:StartClasses"/>
<owl:Class rdf:about="saha:ClassDescription"/>
<owl:Class rdf:about="saha:PropertyDescription"/>
<owl:Class rdf:about="saha:QueryProperties"/>

<owl:ObjectProperty rdf:about="saha:describes">
    <rdfs:domain>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <owl:Class rdf:about="saha:ClassDescription"/>
                <owl:Class rdf:about="saha:PropertyDescription"/>
            </owl:unionOf>
        </owl:Class>
    </rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="saha:annotationProperty">
    <rdfs:domain rdf:resource="saha:ClassDescription"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="saha:annotationClass">
    <rdfs:domain rdf:resource="saha:StartClasses"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="saha:labelProperty">
    <rdfs:domain rdf:resource="saha:ClassDescription"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="saha:onkiURL">
    <rdfs:domain>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <owl:Class rdf:about="saha:ClassDescription"/>
                <owl:Class rdf:about="saha:PropertyDescription"/>
            </owl:unionOf>
        </owl:Class>
    </rdfs:domain>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="saha:allowInstanceValues">
    <rdfs:domain rdf:resource="saha:PropertyDescription"/>
</owl:DatatypeProperty>
```

```
<!-- Aloitusluokat: -->

<saha:StartClasses rdf:ID="start_classes">
  <saha:annotationClass rdf:resource="schema:DocumentType"/>
</saha:StartClasses>

<!-- Asetukset annotaatioskeeman luokille: -->

<saha:ClassDescription rdf:ID="ClassDescription_01">
  <saha:describes rdf:resource="foaf:Person"/>
  <saha:labelProperty rdf:nodeID="labels_01/>
  <saha:annotationProperty rdf:nodeID="properties_01"/>
</saha:ClassDescription>
<rdf:Description rdf:nodeID="labels_01">
  <rdf:type
     rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq"/>
  <rdf:li rdf:resource="foaf:name"/>
  <rdf:li rdf:resource="foaf:mbox"/>
</rdf:Description>
<rdf:Description rdf:nodeID="properties_01">
  <rdf:type
     rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq"/>
  <rdf:li rdf:resource="foaf:name"/>
  <rdf:li rdf:resource="foaf:mbox"/>
</rdf:Description>

<saha:ClassDescription rdf:ID="ClassDescription_02">
  <saha:describes rdf:resource="schema:DocumentType"/>
  <saha:labelProperty rdf:nodeID="labels_02/>
  <saha:annotationProperty rdf:nodeID="properties_02"/>
</saha:ClassDescription>
<rdf:Description rdf:nodeID="labels_02">
  <rdf:type
     rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq"/>
  <rdf:li rdf:resource="dc:title"/>
  <rdf:li rdf:resource="dc:description"/>
</rdf:Description>
<rdf:Description rdf:nodeID="properties_02">
  <rdf:type
     rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq"/>
  <rdf:li rdf:resource="dc:title"/>
  <rdf:li rdf:resource="dc:description"/>
  <rdf:li rdf:resource="dc:creator"/>
</rdf:Description>

<!-- Asetukset annotaatioskeeman ominaisuuksille: -->

<saha:PropertyDescription rdf:ID="d_01">
  <saha:describes rdf:resource="dc:description"/>
  <saha:onkiURL rdf:resource="http://demo.seco.tkk.fi/onki/yso/"/>
  <saha:allowInstanceValues>false</allowInstanceValues>
</saha:PropertyDescription>

</rdf:RDF>
```

LIITE 3. Esimerkki Sahan asetustiedostosta

Alla on esimerkki Sahan asetustiedostosta. Asetustiedosto on tekstitiedosto, jossa määritellään avain-arvo-pareilla tietyyn annotaatioprojektiin liittyvät asetukset.

```
db_user=username
db_password=password
db_host=demo.seco.hut.fi
db_port=5432
db_name=saha
allow_new_subclass=true
annotates_property=http://demo.seco.tkk.fi/saha#annotates
```

LIITE 4. Käyttökokemuskysely, Terveyden edistämisen keskus ry (Tekry)

Saha annotaatioeditori 20.2.2006
Kysymyksiä käyttökokemuksista
FinnONTO/Soster Onni Valkeapää, SeCo

Tällä kyselyllä pyritään kartoittamaan Saha-annotaatioeditorin käyttökokemuksia ja niiden kautta kehittämään sekä parantamaan editoria. Yritä vastata niin moneen kysymykseen kuin mahdollista. Erityisesti silloin, kun vastaus johonkin kysymykseen on kielteinen, erittele tarkemmin mahdollisesti kohtaamiasi ongelmia. Vastaukset ja kyselyä koskevat tiedustelut voi lähettää sähköpostilla osoitteeseen: onni.valkeapaa@tkk.fi. Kyselyn vastaukset käsitellään nimettöminä.

1. Yleistä
    a. Tuntuiko annotointiin liittyvät käsitteet selviltä ja ymmärrettäviltä?
    b. Oliko käyttäjällä selkeä tuntemus siitä, mitä editorilla tehdään ja miksi annotaatioita tuotetaan?
    c. Kokiko käyttäjä tekemänsä työn hyödylliseksi?
    d. Tuliko editoria käyttäessä vastaan jotain sellaisia termejä, jotka eivät olleet tuttuja?
    e. Olivatko editorin käyttöliittymässä esiintyvät termit sujuvan käytön kannalta riittävän selkeitä?
    f. Esiintyikö editorin käytössä teknisiä ongelmia esimerkiksi käytettävän internet-selaimen suhteen? Jos esiintyi, erittele tarkemmin.
    g. Oliko editorin toiminnassa katkoja?
    h. Antoiko editori virheilmoituksia? Jos antoi, erittele tarkemmin.

2. Annotaation avaaminen ja luonti
    a. Oliko käyttäjälle selvää, mitä annotaatioluokalla tarkoitetaan ja mikä on sen yhteys annotaatioon?
    b. Oliko sopivan annotaatioluokan valinta helppoa uutta annotaatiota luotaessa?
    c. Pystyikö käyttäjä helposti näkemään, mitä annotaatioita on aikaisemmin luotu?
    d. Oliko aikaisemmin luotujen annotaatioiden avaaminen helppoa?
    e. Oliko annotaatioiden avaussivu selkeä?
    f. Oliko uuden annotaation luontisivu selkeä?

3. Annotointisivu
    a. Oliko käyttäjän helppo hahmottaa, minkälaisia arvoja eri kenttiin pitää syöttää?
    b. Olivatko eri ominaisuuksille syötetyt arvot näkyvillä riittävän selvästi (taulukot, joissa on keltainen pohja)?
    c. Oliko täytettäviä ominaisuuksia sopivasti vai liian paljon?
    d. Puuttuiko täytettävien ominaisuuksien joukosta jotain oleellista?
    e. Tuntuiko instanssi-tyyppisten arvojen (esim. ”Organisaatio”) syöttäminen selkeältä?

f. Oliko käyttäjän helppo hahmottaa, milloin luotiin uutta instanssia ja milloin valittiin aikaisemmin luotu, olemassa oleva instanssi?
g. Esitettiinkö aikaisemmin luodut instanssit (esim. Henkilöt ja Organisaatiot) ja niiden valintamahdollisuus ominaisuuden arvoksi riittävän selkeästi?
h. Oliko ONKI:sta haettavien arvojen syöttäminen helppoa?
i. Kumpi ONKI-arvojen syöttötavoista tuntui helpommalta: ONKI:n avaaminen uuteen ikkunaan vai hakusanan syöttäminen ja käsitteen valinta hakutuloslaatikosta?
j. Oliko annotaation tallentaminen ja sulkeminen selkeää?
k. Oliko annotointisivu selkeä?

Edellisiin kysymyksiin annettujen vastauksien lisäksi myös kaikki muu Sahaan liittyvä vapaamuotoinen palaute on tervetullutta.

Kiitos vastauksistasi!