

# **Yhteiskäyttöisten käsitteiden kuvauslogiikkaperusteinen määrittely webissä**

Petri Lindgren

Helsinki 25.4.2007

Pro gradu -tutkielma

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Tiedekunta/Osasto — Fakultet/Sektion — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author Petri Lindgren			
Työn nimi — Arbetets titel — Title Yhteiskäyttöisten käsitteiden kuvauslogiikkaperusteinen määrittely webissä			
Oppiaine — Läroämne — Subject			
Työn laji — Arbetets art — Level Pro gradu -tutkielma		Aika — Datum — Month and year 25.4.2007	Sivumäärä — Sidoantal — Number of pages 88 sivua + 14 liitesivua
Tiivistelmä — Referat — Abstract			
<p>Yritystenvälisen kaupankäynnin ja muiden yritysten toisilleen tarjoamien prosessien automatisoinnissa eräs suurimmista haasteista on ollut tietojärjestelmien välisen yhteentoimivuuden toteuttaminen. Näin on ollut jo yli vuosikymmenen. Yhteentoimivuuden toteuttaminen muodostaa tavallisesti 25-70 prosenttia tietojärjestelmien kehitys- ja ylläpitokustannuksista.</p> <p>Puhtaasti syntaktiset menetelmät järjestelmien välisen semanttisen yhteentoimivuuden toteuttamiseksi tekevät liian paljon virheitä. Semanttinen yhteentoimivuus edellyttää eri lähteistä peräisin olevan tiedon merkityksen lähes virheetöntä tunnistamista. Järjestelmien välisen semanttisen yhteentoimivuuden toteuttamisessa tarvitaan siksi sisällönkuvailua käsitteiden avulla.</p> <p>Nykyisten tietojärjestelmissä käytettyjen käsitteistöjen ongelma on se, että vaikka eri käsitteistöt kuvailisivat monelta osin samoja tai osin päällekkäisiä kategorioita, tätä ei pystytä automaattisesti tunnistamaan. Semanttisen yhteentoimivuuden toteuttamiseksi nykyisin käytettyjen käsitteistöjen välille on siksi määriteltävä ihmisvoimin eksplisiittisiä semanttisia suhteita. Tästä aiheutuu merkittävä osa semanttisen yhteentoimivuuden toteuttamisen suurista kustannuksista.</p> <p>Jos toimijat määrittelevät käsitteensä kuvauslogiikalla, käyttäen samaa primitiivikäsitteistöä tai primitiivikäsitteistöä, josta on määritelty merkitysvastaavuussuhteet toisten käyttämiin primitiivikäsitteistöihin, niin toimijoiden määrittelemien käsitteiden väliset merkitysvastaavuussuhteet voidaan löytää automaattisesti. Tällä kuvauslogiikka+primitiivikäsitteistö-menetelmällä toimijat voivat määrittellä tarvitsemiaan käsitteitä toisistaan riippumattomasti, ja saavuttaa semanttisen yhtenäisyyden.</p> <p>Tässä tutkielmassa tarkastellaan kuvauslogiikka+primitiivikäsitteistö-menetelmän taustalla olevaa teoriaa, menetelmän soveltamista käytännön esimerkkien avulla ja menetelmän tuomia etuja ja sen soveltamisen asettamia haasteita.</p> <p>ACM Computing Classification System (CCS):  I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods  F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic  H.3.5 [Information Storage and Retrieval]: Online Information Services</p>			
Avainsanat — Nyckelord — Keywords Yhteentoimivuus, Kuvauslogiikka, OWL, SUMO, Ontologia, Semanttinen web			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — övriga uppgifter — Additional information			

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Tietämyksen yhteiskäyttöisyyden ongelmia</b>	<b>4</b>
2.1	Vapaan tekstikuvailun ongelmia . . . . .	4
2.2	Käsiteperustaisten kuvailujen ongelmia . . . . .	6
<b>3</b>	<b>Teorioita ja tietomalleja ihmisen muistin rakenteesta</b>	<b>9</b>
3.1	Klassinen teoria käsitteistä . . . . .	9
3.1.1	Semanttiset verkot -tietomalli . . . . .	10
3.1.2	Kehystietomalli . . . . .	13
3.2	Prototyypiteoria ja konnektionistinen tietomalli . . . . .	13
<b>4</b>	<b>Kuvauslogiikka semanttisen yhteentoimivuuden perustana</b>	<b>17</b>
4.1	Käsitteiden määrittely . . . . .	17
4.2	Käsitteiden automaattinen tulkinta . . . . .	22
4.3	Esimerkki kuvauslogiikan soveltamisesta . . . . .	25
<b>5</b>	<b>Yläontologia primitiivikäsitteistönä</b>	<b>28</b>
5.1	Perustason käsitteet . . . . .	28
5.2	Primitiivikäsitteet . . . . .	29
5.3	Yläontologiat primitiivikäsitteistönä . . . . .	31
<b>6</b>	<b>Kuvauslogiikka käytännön web-sovelluksissa</b>	<b>34</b>
6.1	OWL-kuvauslogiikka . . . . .	35
6.2	Päättekoneita OWL-kuvauslogiikalle . . . . .	38
<b>7</b>	<b>Kuvauslogiikka+primitiivikäsitteistö-menetelmän soveltaminen</b>	<b>40</b>

7.1	Käsitteen palvelu merkitys . . . . .	40
7.2	STAKES-palveluluokitus ja sen rajoitukset . . . . .	41
7.3	Määriteltävät palvelukäsitteet . . . . .	43
7.4	Palvelukäsitteiden määrittely OWL+SUMO-menetelmällä . . . . .	44
7.5	Käsitteiden väliset automaattisesti pääteltävät suhteet . . . . .	63
<b>8</b>	<b>Kuvauslogiikka+primitiivikäsitteistö-menetelmän arviointi</b>	<b>67</b>
8.1	OWL+SUMO-menetelmän ilmaisuvoima . . . . .	67
8.2	Kuvauslogiikka+primitiivikäsitteistö-menetelmän hyötyjä . . . . .	69
8.3	OWL+primitiivikäsitteistö-menetelmän soveltamishaasteita . . . . .	71
<b>9</b>	<b>Vastaavia menetelmiä</b>	<b>75</b>
9.1	GALEN-projekti . . . . .	75
9.2	ISO 15926 -standardi . . . . .	76
<b>10</b>	<b>Yhteenveto</b>	<b>78</b>
	<b>Lähteet</b>	<b>80</b>
	<b>Liitteet</b>	
	<b>1 Esimerkkikäsitteiden määrittelyt sarjallistettuna XML-esitysmuotoon</b>	
	<b>2 Toimijan määrittelemät kyselykäsitteet sarjallistettuna XML-esitysmuotoon</b>	

# 1 Johdanto

Tietojärjestelmien välisen yhteentoimivuuden (engl. system interoperability) toteuttaminen on ollut eräs suurimmista haasteista jo yli vuosikymmenen. Erityisesti näin on ollut yritystenvälisessä kaupankäynnissä ja muissa yritysten toisilleen tarjoamissa palveluissa [GLP05, PR04, Wes03]. Palveluiden kysynnän kasvun ja monipuolistumisen myötä tietojärjestelmien välisen yhteentoimivuuden toteuttamisesta on kehitymässä entistäkin suurempi haaste taloudelliselle kasvulle.

Kehittyneiden teollisuusmaiden bruttokansantuotteesta jo yli 60 prosenttia muodostuu palveluista. Palveluiden, erityisesti tietämyspalveluiden<sup>1</sup> (engl. knowledge-based services), suhteellinen osuus kasvaa edelleen muiden toimintojen muodostamien suhteellisten osuuksien pienentyessä. Esimerkiksi taloudellisen yhteistyön ja kehityksen järjestöön OECD kuuluvissa maissa tehdasteollisuuden muodostama osuus bruttokansantuotteesta on pienentynyt alle 20 prosentin. Kehittyneissä teollisuusmaissa palveluista on tulossa tai on jo tullut tuottavuuden ja taloudellisen kasvun perusta [GLP05, OST00].

Erityisesti internetperustaiset sovellukset ovat osoittautuneet kustannustehokkaiksi automatisoitaessa erilaisia yritystoimintaan liittyviä hajautettuja prosesseja. Osoituksena tästä on esimerkiksi se, että vuonna 2003 yli 90 prosenttia yrityksistä kehittyneissä länsimaissa käytti internetiä toiminnassaan [kan06, GLP05]. Suomessa 65 prosentilla yrityksistä on kotisivut. Näistä yrityksistä 24 prosenttia harjoitti myyntiä internetissä vuonna 2004. Tällä hetkellä internet-myynti on nopeassa kasvussa. Nopeimmin se on kasvanut yritystenvälisessä (B2B) myynnissä. B2B-myyntin osuus kaikkien asiakasryhmien välisestä myynnistä on 86 prosenttia [kan06]. Tietojärjestelmien muokkaamisesta yhteentoimiviksi eli tietojärjestelmien yhteenliittämisestä (engl. system integration) onkin tullut nopeimmin kasvava osa tietoteollisuutta. Siitä on myös tullut runsaasti tarpeettomia kustannuksia aiheuttava toimenpide. [GLP05, PR04, Wes03].

Tietojärjestelmien välinen yhteentoimivuus on ”kahden tai useamman järjestelmän tai komponentin kyky vaihtaa tietoa ja käyttää tätä vaihdettua tietoa” [IEE90]. Yh-

---

<sup>1</sup>Tietämys on asiantuntijatietoa. Toisin sanoen tietämys ei ole mitä tahansa tietoa vaan jonkin aihealueen tai aihealueiden monitahoisemmiksi kokonaisuuksiksi yhdisteltyä tietoa [Nii96].

teentoimivuus voidaan jakaa *syntaktiseen* ja *semanttiseen yhteentoimivuuteen*. Tietojärjestelmät ovat syntaktisesti yhteentoimivia, jos ne pystyvät vaihtamaan tietoa keskenään. Tämän mahdollistavat yhteisesti sovitut rajapinnat ja tiedon esitysmuodot esimerkiksi World Wide Web (web) ja XML-esitysmuoto<sup>2</sup> [Hor02]. Huomattavasti vaikeampi ongelma on semanttinen yhteentoimivuus [AG06]. Tietojärjestelmät ovat semanttisesti yhteentoimivia, jos ne pystyvät käyttämään tarkoituksenmukaisesti keskenään vaihtamaansa tietoa. Semanttinen yhteentoimivuus on pohjimmiltaan sitä, että yhteentoimivat tahot tuntevat tai tunnistavat toistensa keskenään vaihtaman tiedon tarkoitetun merkityksen [PR04, Wes96]. *Tiedon merkityksen tunnistaminen* on taas pohjimmiltaan sitä, että tunnistetaan miten ”uusi” tieto suhteutuu ”vanhaan”, merkitykseltään jo tunnettuun tietoon [MR05].

Yhteentoimivuuden toteuttaminen muodostaa tavallisesti 25-70 prosenttia tietojärjestelmien kehitys- ja ylläpitokustannuksista [Wes96]. Merkittävä syy näille kustannuksille on tietojärjestelmien sisältämän tiedon merkityksen joustamaton ja puutteellinen mallinnus, joka ei mahdollista eri lähteistä peräisin olevan tiedon merkityksen automaattista ja luotettavaa tunnistamista [Wes96, NP01]. Jos eri lähteistä peräisin olevan tiedon merkitystä ei pystytä tunnistamaan automaattisesti ja luotettavasti, niin semanttinen yhteentoimivuuden aikaansaamiseksi jokaisen tietojärjestelmän *A* kannalta kiinnostavan ja asiaankuuluvan tietolähteen *B* tiedot joudutaan yhdistämään tapauskohtaisesti ja ihmisvoimin järjestelmän *A* tietoihin. Tämä on työlästä ja kallista [Wes96, BCM<sup>+</sup>03a].

Semanttisen yhteentoimivuuden toteuttaminen ei ole ongelma ainoastaan yritysten tai muiden toimijoiden välisissä prosesseissa vaan myös toimijoiden itsensä tuottaman tiedon uudelleen hyödyntämisessä. Esimerkiksi erilaisissa organisaatioissa on monesti ongelmana se, että saamaa tietoa tuotetaan eri tarkoituksia varten eri näkökulmista yhä uudelleen, koska ei pystytä automaattisesti tunnistamaan sitä, miten jo aiemmin tuotettu ”vanha” tieto suhteutuu ”uuteen” tietoon ja käyttötarkoitukseen [Wes96].

Tämän tutkielman tavoitteena on esitellä menetelmä, jolla toimijat voivat määritellä toisistaan riippumattomasti tietojärjestelmissään käsiteltävän tiedon merkityksen

---

<sup>2</sup><http://www.w3.org/XML/>

niin, että tämä tieto on automaattisesti tunnistettavissa eli suhteutettavissa muuhun samalla menetelmällä kuvailtuun tietoon. Esimerkkinä tarkastellaan terveydenhuollon palveluihin liittyvän tiedon merkityksen määrittelyä ja automaattista tunnistamista.

## 2 Tietämyksen yhteiskäyttöisyyden ongelmia

Nykyisissä tietojärjestelmissä tiedon merkitys on pääsääntöisesti esitetty vapailla tekstikuvauksilla tai käsitteillä. Vapaat tekstikuvaukset ovat täsmällisemmin sanottuna kielellisiä ilmauksia eli *termejä* tai *lauseita*. Termit ovat tekstin sanoja tai niiden muodostamia jäsentyneitä kokonaisuuksia: *yksilötermejä* tai *predikaatteja*. Yksilötermejä ovat esimerkiksi ”Suomen nykyinen presidentti” tai kaikki erisnimet. Yksilötermit nimeävät kohteita (olioita). Predikaateilla ilmaistaan oliolla olevia suhteita tai ominaisuuksia. Esimerkiksi ”punainen” ja ”pienempi kuin” ovat predikaatteja. Lauseiden avulla voidaan ilmaista tosia tai epätosia väittämiä, kuten ”palvelu on prosessi” [Nii97].

Käsitteet ovat termien merkityksiä (intensioita) eli kuvailuja termien viittauskohteista (ekstensiosta). Termin ekstensio on niiden olioiden tai suhteiden joukko, johon intensio pätee. Toisin sanoen termin intensio määrää sen ekstension. Termin ekstensioon kuuluvilla olioilla (tai suhteilla) on yksi tai useampi yhteinen piirre, jota muilla olioilla (tai suhteilla) ei ole. Esimerkiksi termin *kissa* merkitys eli käsite *kissa* kuvailee kaikki vain kissoille tyypilliset piirteet. Joukkoa joiden jäsenillä on yksi tai useampi yhteinen piirre, jota muilla olioilla ei ole kutsutaan *kategoriaksi* [Mur02a, Nii97, MR05].

### 2.1 Vapaan tekstikuvailun ongelmia

Ihanteellista olisi, jos nykyisillä tietoteknisillä sovelluksilla pystyttäisiin tunnistamaan kielellisten ilmausten merkityksiä ihmisten tavoin. Tällöin semanttinen yhteentoimivuus olisi mahdollista toteuttaa ihmiselle luontevan tietämyksen esitystavan varaan. Kielellisten ilmausten välisten merkitysvastaavuuksien löytäminen automaattisesti on kuitenkin hyvin ongelmallista.

*Vapaan tekstihaun* (engl. free search text tai full text search) avulla käyttäjän on mahdollista löytää semanttisia vastaavuuksia kielellisten ilmausten välillä. Se palauttaa käyttäjällensä tekstit, joissa esiintyy samoja sanoja kuin käyttäjän hakutekstissä. Tämä ei kuitenkaan vielä riitä kielellisten ilmausten välisten semanttisten vastaavuuksien virheettömään löytämiseen. Vapaan tekstihaun avulla saadaan har-



voin noudettua yli 20 prosenttia tietokannan hakukohteista, jotka ovat todennäköisesti haun suhteen relevantteja [TL01].

Monet sanat ovat merkitykseltään epämääräisiä tai monimerkityksellisiä. Luonnollisessa kielessä samaan merkitykseen voidaan viitata useilla eri synonyymeilla ja toisaalta yksi sana voi viitata useaan eri merkitykseen<sup>3</sup>. Vapaisiin tekstikuvailuihin ja -hakuihin perustuvien menetelmien hakutulosten saanti ja erityisesti tarkkuus ovat huonoja, koska sanat eivät yksilöi merkityksiä. Vapaan tekstihaun hakutulos sisältää tyypillisesti myös tekstejä, jotka eivät vastaa semanttisesti hakutekstiä. Esimerkiksi kaksi erillistä kielellistä ilmausta eivät välttämättä vastaa tai edes liity semanttisesti toisiinsa, vaikka niissä esiintyy samoja sanoja. Toisaalta vaikka kahdesta kielellisestä ilmauksesta ei löytyisikään samoja sanoja, niin ne voivat vastata semanttisesti toisiaan. Niinpä vapaan tekstihaun avulla löydetty merkitysvastaavuussuhteet voivat sisältää virheellisiä suhteita tai ne voivat olla puutteellisia.

Vapaiden tekstihakujen tarkkuutta huonontaa lisäksi niiden kyvyttömyys tai virhealtis kyky tunnistaa kahden tai useamman sanan muodostamia termejä ja lauseita, jotka viittaavat tiettyyn merkitykseen. Kielellisellä analyysillä (engl. Natural Language Processing) (NLP) näitä termejä ja lauseita on mahdollista tunnistaa jäsentämällä kielellisiä ilmauksia, mutta NLP-menetelmät ovat tältä osin alttiita virheille [TL01, Hov05, Lei03]. NLP-menetelmillä pystytään jäsentämään oikein esimerkiksi vain noin 30 prosenttia lääketieteellisistä teksteistä [BC02].

Virheitä aiheuttavat erityisesti NLP-menetelmien vaikeudet tunnistaa pääsana ja sitä tarkentava sana tai lause. Tämän tunnistamiseksi tarvitaan monesti *yleistä semanttista tietämystä*, jota syntaksissa ei ole. Tällainen yleinen semanttinen tietämys on ihmisten yleistietoa esimerkiksi historiasta, kulttuurista ja siitä miten asiat maailmassa yleisesti ovat [TL01, Hov05, Lei03]. Jos kielellisen ilmauksen tulkitsemiseen tarvitaan syntaksin lisäksi yleistä semanttista tietämystä, niin NLP-menetelmistä ei ole juurikaan apua [TL01, Hov05, Lei03].

Vapaisiin tekstihakuihin liittyvät ongelmat on helppo todeta esimerkiksi olemassa olevilla palvelunhakujärjestelmillä. Suomessa paljon käytetyt palvelunhakujärjes-

---

<sup>3</sup>Monimerkityksellisiä sanoja kutsutaan *homonyymeiksi*.

telmät ovat muiden muassa Fonectan 020202.fi<sup>4</sup> ja Eniron yritykset.enero.fi<sup>5</sup>. Jos etsimme näiden järjestelmien termihauilla tietokoneen korjauspalveluita<sup>6</sup> termeillä tietokonekorjaus ja tietokoneen korjaus, niin Fonectan järjestelmä ei palauta yhtäkään palveluilmoitusta vaikka kyseisen järjestelmän tietämuskannassa on tietokoneen korjauspalveluiden ilmoituksia. Näitä löytyy esimerkiksi hakutermillä tietokonehuolto.

Tämä ongelma seuraa todennäköisimmin siitä, että Fonectan järjestelmän termihaku ei osaa päätellä tai sillä ei ole tietoa siitä, että käyttäjän käyttämä termi tietokonekorjaus ja kahden termin muodostama kokonaisuus tietokoneen korjaus vastaa merkitykseltään palveluilmoituksen laatijan käyttämän termin tietokonehuolto merkitystä. Termihaku ei siis osaa päätellä tai sillä ole tietoa siitä, että nämä termit (tai termien muodostamat kokonaisuudet) ovat synonyymeja. Eniron palvelunhakujärjestelmässä on samanlaisia ongelmia. Hakutermillä tietokoneen korjaus se ei palauta yhtään ilmoitusta, kun taas termillä tietokonekorjaus se palauttaa yhden ilmoituksen tietokoneen korjaus -palvelusta. Hakutermillä tietokonehuolto Eniron järjestelmästä löytyy yli 200 ilmoitusta. Ihmiset tekevät myös paljon kirjoitusvirheitä, mikä aiheuttaa lisäongelmia vapaisiin tekstihakuihin.

Semanttinen yhteentoimivuus edellyttää lähes virheetöntä merkitysten tunnistamista. Jos tietämys suhteutetaan väärin jo tunnettuun tietoon, niin se johtaa todennäköisesti virheelliseen toimintaan, jossa vaihdettua tietoa ei hyödynnetä tarkoitettulla tavalla. Vaikka toimijoiden välisen semanttisen yhteentoimivuuden ei tarvitse olla täydellistä hyödyllisen yhteistyön aikaansaamiseksi, niin puhtaasti syntaktiset menetelmät tekevät kuitenkin liian paljon virheitä. Järjestelmien välisen semanttisen yhteentoimivuuden toteuttamisessa tarvitaan siksi sisällönkuvailua käsitteiden avulla [Rog06, Wes03, TL01].

---

<sup>4</sup><http://020202.fi>

<sup>5</sup><http://yritykset.enero.fi>

<sup>6</sup>Testihaut on tehty 1.10.2006.

## 2.2 Käsiteperustaisten kuvailujen ongelmia

Tietojärjestelmissä käsitteellä tarkoitetaan käytännössä termille (tai jollekin vastaavalle syntaktiselle tunnisteelle) sovittua ja määriteltyä merkitystä [KS05]. Semanttisen yhteentoimivuuden ongelmaa ei olisi, jos kaikki toimijat kuvailisivat tiedon merkityksen tällaisilla yhteisesti sovituilla käsitteillä. Ei kuitenkaan ole realistista olettaa, että kaikilla mahdollisesti toistensa palveluita hyödyntävillä toimijoilla olisi yhteisesti sovittuna yksi valtava ajantasainen käsitteistö, joka sisältäisi käsitteet kuvaamaan esimerkiksi jokaisen toimijan tarjoamaa palvelutyyppiä [FBS02]. Toimijoiden on pystyttävä määrittelemään käsitteitä omien tarpeidensa mukaan, muista toimijoista riippumattomasti. Näin määriteltyjen käsitteiden muodostamaan kokonaisuuteen viitataan tässä tutkielmassa nimellä *toimijakohtainen käsitteistö*.

Nykyisten tietojärjestelmissä käytettyjen toimijakohtaisten käsitteistöjen ongelma on se, että yhden toimijan käyttämät kategorioita edustavat tunnisteet eivät liity semanttisesti toisen toimijan käyttämiin tunnisteisiin tai tätä liittymistä ei pystytä automaattisesti päättämään vaikka tunnisteet viittaisivat samaan tai osin päällekkäiseen kategoriaan. Toisin sanoen vaikka eri toimijakohtaiset käsitteistöt kuvailisivat monelta osin samoja tai osin päällekkäisiä kategorioita, tätä ei pystytä automaattisesti tunnistamaan. Semanttisen yhteentoimivuuden toteuttamiseksi nykyisten toimijakohtaisten käsitteistöjen välille on siksi määriteltävä ihmisvoimin eksplisiittisiä semanttisia suhteita. Tästä aiheutuu merkittävä osa semanttisen yhteentoimivuuden toteuttamisen suurista kustannuksista.

Jos toimijakohtaisten käsitteistöjen välille joudutaan määrittelemään semanttisia suhteita ihmisvoimin, niin on käytännössä mahdotonta saavuttaa tilanne, jossa tietojärjestelmästä on semanttinen yhteentoimivuus mahdollisimman kattavaan joukkoon asiaankuuluvia tietämuskantoja. Toimijan tarpeet tyydyttäviä palveluita tai niihin liittyvää ja tarvittavaa tietämystä voi olla tarjolla useissa eri tietämuskannoissa useiden eri toimijoiden tietojärjestelmissä, jolloin mahdollisten semanttisten suhteiden määrä voi nousta nopeasti hyvin suureksi. Myös käsitteistöissä jatkuvasti tapahtuvat muutokset lisääisivät entisestään ihmisvoimin toteutettavan semanttisen yhteentoimivuuden työläyttä. Toimijakohtaisten käsitteistöjen väliset semanttiset suhteet on pystyttävä löytämään automaattisesti.

Ratkaisua käsitteiden automaattiseen tulkintaan on etsitty muiden muassa tutkimalla ja muodostamalla erilaisia teorioita ihmisen muistin rakenteesta ja pyrkimällä toteuttamaan näitä teorioita vastaavia laskennallisia malleja. Tässä tutkielmassa tarkastelemme erityisesti klassista teoriaa käsitteistä ja siitä johdettuja laskennallisia malleja.

Klassinen teoria on yksi ensimmäisistä teorioista, joilla on pyritty selittämään ihmisen muistin rakennetta ja käsitteisiin liittyvää merkitystä eli sitä miten käsitteet kuvailevat kategorioita. Nykyiset voimakkaasti esillä olevat mallit käsitteiden esittämiseksi siten, että niiden avulla pystyttäisiin algoritmisesti selvittämään käsitteisiin liittyvä merkitys, on johdettu ja kehitelty tästä klassisesta teoriasta.

### 3 Teorioita ja tietomalleja ihmisen muistin rakenteesta

Ihmisen mielessä käsitteet ovat kategorioiden kuvailuja. Käsitteiden kuvailemat kategoriat koostuvat yksilöistä, joita ihmiset yleisesti pitävät samanlaisina jostakin tietystä näkökulmasta tarkasteltuna. Yleisesti ajatellaan, että yksilöt ovat keskenään samanlaisia, jos niillä on yksi tai useampi yhteinen piirre, jota muilla yksilöillä ei ole tai joilla esiintyy tilastollisesti useammin yksi tai useampi yhteinen piirre, kuin muilla yksilöillä [MR05].

Prosessia, joka määrittelee tietyn olion kuuluvaksi tiettyyn kategoriaan, kutsutaan *kategorisoinniksi* [MR05]. Kategorisoinnin avulla ihmiset voivat liittää käsitteiden kuvailujen sisältämää tietämystä ennestään tuntemattomiin yksilöihin. Jos yksilö  $y$  on jostakin näkökulmasta tarkasteltuna samanlainen kategorian  $K$  yksilöiden kanssa, niin se luokitellaan kuuluvaksi kategoriaan  $K$ . Tämän jälkeen kategoriaa  $K$  edustavaan käsitteeseen liittyvä tietämys on myös tietämystä yksilöstä  $y$ . Esimerkiksi jos ihminen kategorisoi jonkin olion taskulampuksi, niin hän voi melko varmasti päätellä, että tässä oliossa on kytkin, jonka avulla se saadaan tuottamaan valoa [MR05]. Kategorisointi on keino ymmärtää, ennustaa ja hallita mahdollisimman tehokkaasti monimutkaista ympäristöä [MR05, GS05].

#### 3.1 Klassinen teoria käsitteistä

Käsitteitä selittävän *klassisen teorian* [Mur02a] perusteella ihmisen mielessä käsitteitä edustavat ja niiden merkitystä kuvailevat riittävistä ja välttämättömistä kategoriaan kuulumisen ehdoista koostuvat määritelmät. *Välttämättömät* ehdot määrittelevät käsitteen ne ehdot, joiden on täytyttävä, jotta yksilö voi kuulua käsitteen edustamaan kategoriaan. Jos yksilö ei täytä käsitteen  $C$  välttämättömiä ehtoja, niin se ei voi kuulua käsitteen  $C$  kuvailemaan kategoriaan. Käsitteen *välttämättömät ja riittävät ehdot* sisältävät sen täydellisen määritelmän. Jos yksilö täyttää käsitteen  $C$  välttämättömät ja riittävät ehdot niin yksilön täytyy kuulu siihen kategoriaan, jota käsite  $C$  kuvailee. Klassinen teoria väittää lisäksi, että yksilöt joko kuuluvat tai eivät kuulu johonkin tiettyyn kategoriaan. Epätasällisyyttä tämän suhteen ei

sallita [Mur02a].

### 3.1.1 Semanttiset verkot -tietomalli

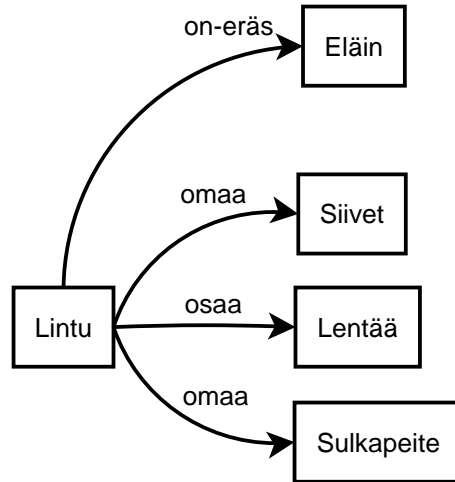
Quillian (1967) esitti Semantic Networks (semanttiset verkot) -nimisen tietomallin, jossa tietämyksen ajatellaan olevan talletettuna ihmisen muistiin käsitteiden ja yksilökuvausten välisiä suhteita vastaavalla tavalla. Tällainen tietomalli voidaan kuvata verkkona, jonka solmut vastaavat käsitteitä tai yksilökuvauksia ja suunnatut kaaret niiden välisiä suhteita. Semanttisten verkkojen merkintätavassa määriteltävään käsitteeseen tai yksilökuvaukseen viitataan tavallisesti nimellä *subjekti*, käsitteiden tai yksilökuvausten välisiin suhteisiin viitataan nimellä *predikaatti* ja niihin käsitteisiin tai yksilökuvauksiin, joilla on jokin suhde subjektiin viitataan nimellä *objekti* [MR05, Woo75].

Oletetaan esimerkiksi, että käsitettä *Lintu* edustaisi mielessä määritelmä: Kategoriiaan lintu kuulumisen välttämätön ehto on se, että yksilö on eläin. Jos sillä on tämän lisäksi siivet, se osaa lentää ja sillä on sulkapeite, niin yksilö täyttää välttämättömät ja riittävät ehdot kuuluakseen kategoriiaan lintu. Semanttisten verkkojen teorian perusteella tämä määritelmä on talletettuna ihmisen muistiin kuvassa 1 esitettyä verkkoa vastaavalla rakenteella [MR05]. Siinä subjektina on *Lintu*. Predikaatteja edustavat suunnatut nuolet. Kolmikot tulkitaan suoraviivaisesti, esimerkiksi *Lintu-omaa-Sulkapeite-kolmikon* tulkinta olisi ”lintu omaa sulkapeitteen”.

Semanttisissa verkoissa kategoriiaan kuulumisen välttämättömät ehdot osoitetaan predikaatilla *on-eräs* (engl. ”is a” tai ”a kind of”) (*is-a*). Muut predikaatit yhdessä *is-a*-predikaatin kanssa ilmaisevat kategoriiaan kuulumisen riittävät ehdot. Käsitteen *C* välttämättömät ehdot ovat käsitteen *C* välttämättömien ja riittävien ehtojen muodostaman joukon osajoukko. Niinpä käsitteen *C* välttämättömien ja riittävien ehtojen määrittelemä yksilöiden joukko  $J_C$  on käsitteen *C* välttämättömien ehtojen esimerkiksi käsitteen *D* määrittelemän yksilöiden joukon  $J_D$  osajoukko:

$$C \text{ is-a } D \Rightarrow J_C \subseteq J_D \Leftrightarrow \forall j(j \in J_C \Rightarrow j \in J_D)$$

Joukon  $J_D$  tulkinta ( $\mathcal{I}$ ) on ”kaikkien yksilöiden muodostamasta joukosta *E*, niiden yksilöiden muodostama osajoukko, joilla on välttämättömiä ehtoja vastaavat piirteet



Kuva 1: Käsitteen **Lintu** määrittelyä esitettyä semanttisella verkolla.

$v_1, v_2, \dots$  ja  $v_n$ ”:

$$J_D^I = \{j \in E \mid v_1(j), v_2(j), \dots, v_n(j)\}$$

Joukon  $J_C$  tulkinta on ”kaikkien yksilöiden muodostamasta joukosta  $E$ , niiden yksilöiden muodostama osajoukko, joilla on välttämättömiä ehtoja vastaavat piirteet  $v_1, v_2, \dots$  ja  $v_n$ , ja riittäviä ehtoja vastaavat piirteet  $r_1, r_2, \dots$  ja  $r_n$ ”

$$J_C^I = \{k \in E \mid v_1(k), v_2(k), \dots, v_n(k), r_1(k), r_2(k), \dots, r_n(k)\}.$$

**Is-a**-predikaatin ilmaisema käsitettä vastaavien ehtojen osajoukkosuhte on *transiitiivinen*. Jos oletetaan, että  $C$ ,  $D$  ja  $E$  ovat kategorioita kuvaavia käsitteitä, niin voimme todeta yleisemmin

$$(C \text{ is-a } D) \wedge (D \text{ is-a } E) \Rightarrow (C \text{ is-a } E)$$

eli jos  $C \text{ is-a } D$  ja  $D \text{ is-a } E$ , niin tästä seuraa, että  $C \text{ is-a } E$ . Tämä mahdollistaa yksinkertaisen deduktiivisen päättelyn **is-a**-predikaatin suhteen

$$\text{Oletuksista päätellään johtopäätös: } \frac{C \text{ is-a } D \quad D \text{ is-a } E}{C \text{ is-a } E}$$

Is-a-predikaatilla toisiinsa kytkeytyvien käsitteiden muodostamaa kokonaisuutta kutsutaan *hierarkiaksi*. Määrittelyssä  $C$  is-a  $D$ , käsitettä  $C$  kutsutaan käsitteen  $D$  *alakäsitteeksi* ja käsitettä  $D$  käsitteen  $C$  *yläkäsitteeksi*. Vastaavasti käsitteen  $C$  kuvailemaa kategoriaa  $K_C$  kutsutaan käsitteen  $D$  kuvaileman kategorian  $K_D$  *alakatégoriaksi* ja kategoriaa  $K_D$  kutsutaan kategorian  $K_C$  *yläkatégoriaksi*.

Koska kategoriat määräytyvät käsitteiden sisältämien yksilöiltä vaadittavien piirteiden kautta ja transitiivisellä is-a predikaatilla voidaan kertoa subjektikategorialla olevan välttämättöminä piirteinä kaikki objektikategorian piirteet, niin myös seuraava päättely on pätevä

$$\frac{x \in K_C \Rightarrow p_1(x) \wedge p_2(x) \wedge \dots \wedge p_n(x)}{C \text{ is-a } D} \\ y \in K_D \Rightarrow p_1(y) \wedge p_2(y) \wedge \dots \wedge p_n(y)$$

Tämä tarkoittaa, että jos yksilö  $x$  kuuluu käsitteen  $C$  määrittelemään kategoriaan  $K_C$ , niin yksilöllä  $x$  on kaikki kategoriaan  $K_C$  kuulumiseen vaadittavat piirteet  $p_n$ . Jos lisäksi kategorian  $K_C$  määrittelevä käsite  $C$  on käsitteen  $D$  alakäsite ( $C$  is-a  $D$ ) ja on olemassa yksilö  $y$ , joka kuuluu käsitteen  $D$  määrittelemään kategoriaan  $K_D$ , niin yksilöllä  $y$  on vähintään kaikki kategoriaan  $K_C$  kuulumiseen vaadittavat piirteet  $p_n$ . Tätä kutsutaan *piirteiden periytymiseksi* [Mur02a, MR05].

Klassisen teorian perusteella käsitteiden merkitykset ovat samat (tai vastaavat), jos niiden välttämättömät ja riittävät kategoriaan kuulumisen ehdot ovat samat. Semanttisilla verkoilla käsitteiden riittävät ehdot voidaan esittää formaalisti, joten niillä määritellyt merkitykseltään vastaavat käsitteet on mahdollista tunnistaa automaattisesti.

Semanttisten verkkojen algoritmisen soveltamisen ongelmana on ollut erilaiset tulkinnat, joita sen subjektille, objektille ja is-a-predikaatille on tehty. Tämä on seurausta Quillianin esittämän alkuperäisen semanttisten verkkojen määrittelyn epätarkkuudesta. Käytännön sovelluksissa sama semanttinen verkko saatettiin tulkita eri tavoin, mikä teki merkitysvastaavuuksien löytämisestä ja yleisemmin sovellusten yhteentoimivuudesta ongelmallista [Woo75, Mur02a, BCM<sup>+</sup>03b]. Edellä esitetty tulkinta is-a-predikaatille on muodostettu siten, että sen avulla on mahdollista määritellä käsitteitä noudattaen klassista teoriaa.



### 3.1.2 Kehystietomalli

Minsky (1974) vastusti semanttisten verkkojen esittämään tietomalliin hyvin läheisesti liittynyttä ajatusta, jonka mukaan tietämys on esitettyä ihmisen muistiin pirstaleisesti, pieninä palasina. Niinpä hän kehitti *kehystietomallin* [Min74], jolla tietämystä voidaan esittää selkeämmin järjestettynä kokonaisuutena ja paremmin ongelmanratkaisua edistävänä rakenteena. Kehystietomalleissa käsite määritellään joukkona ominaisuuksia (kehysten yhteydessä engl. slot), jotka kuvailevat käsitettä vastaavaan kategoriaan kuuluvien yksilöiden erottelevimpia ominaisuuksia. Ominaisuuksien mahdollisten arvojen joukkoa voidaan rajata ja niille voidaan määritellä oletusarvoja [LM01, Min74]. Niinpä kehystietomallilla esitetyt käsitteiden määritelmät eli *kehykset* ovat rakenteisempia kokonaisuuksia, kuin semanttisilla verkoilla alun perin esitetyt määritelmät [LM01, Min74].

Kehysten välille voidaan määritellä semanttisten verkkojen *is-a*-predikaattia vastaavia suhteita. Niillä voidaan kertoa kehysten olevan toisen kehysten alakehys. Toisin sanoen voidaan kertoa, että alakehykseksi määritelty kehys perii yläkehysten ominaisuudet ja niille mahdolliset määritellyt arvot. Pohjimmiltaan semanttisten verkkojen ja kehysten ero on niin pieni, että semanttiset verkot voidaan toteuttaa kehyksillä ja kehykset semanttisilla verkoilla. Myös kehystietomallin alkuperäinen määrittely oli epätarkka, mikä on hankaloittanut kehysten soveltamista tietämyksen esittämisessä [Sow00, LM01].

## 3.2 Prototyypiteoria ja konnektionistinen tietomalli

Klassisen teorian keskeisin väite on, että yksilö joko kuuluu tai ei kuulu johonkin kategoriaan. Jos yksilöllä  $y$  voidaan havaita olevan kategoriaan  $K$  kuulumiseen vaadittavat piirteet, niin yksilö kuuluu tähän kategoriaan. Ihmisille ei kuitenkaan monestikaan ole täysin selvää mihin kategoriaan jokin yksilö kuuluu. Ovatko esimerkiksi tomaatit vihanneksia vai hedelmiä? Silloinkin kun ihmiselle on selvää mihin kategoriaan yksilö kuuluu, niin voi olla hyvin vaikeata määritellä kategoriaan kuulumiselle vaadittavat välttämättömät ja riittävät ehdot. Miten esimerkiksi käsite *Peli* tulisi määritellä tai *Koira*? Vaikuttaa sille, että aina on mahdollista löytää poikkeuksia,

joilla ei ole kaikkia tiettyyn kategoriaan kuulumiseen vaadittavia piirteitä, mutta jotka kuitenkin ihmisten mielestä selvästi kuuluvat tähän kategoriaan. Jos vaadittavien piirteiden määrää vähennetään tai lisätään siten, että poikkeava yksilö täyttää kategoriaan kuulumiseen vaadittavat ehdot, niin tämän jälkeen uudet muutetut ehdot täyttävien yksilöiden joukossa on poikkeavan yksilön lisäksi sellaisia yksilöitä, jotka eivät sinne ihmisten mielestä kuulu. Ihmisen muistissa kategoriaan kuulumiseen liittyy selvästikin epävarmuutta ja riippumattomuutta yksittäisistä piirteistä [Mur02b, Mur02a].

Käsitteisiin liittyvää epävarmuutta on pystytty tutkimuksilla todistamaan. Esimerkiksi Rosch (1975) osoitti tutkimuksillaan, että ihmiset pitävät joitakin kategoriaan kuuluvia yksilöitä parempina esimerkkeinä kategoriasta, kuin toisia. Tätä taipumusta kutsutaan *prototyypivaikutukseksi*. Ihmiset ovat esimerkiksi sitä mieltä, että lohi on tyypillisempi kala kuin ankerias. Prototyypivaikutustakin vakuuttavampaa osoitusta käsitteiden sumeudesta löysivät tutkimuksillaan muiden muassa McCloskey ja Glucksberg (1978). He osoittivat että tapauksissa, joissa ihmiselle ei ole täysin selvää mihin kategoriaan jokin kohde kuuluu, niin hän saattaa kategorisoida kyseisen kohteen eri tavalla lyhyenkin ajan sisällä [Mur02a, MR05]. Koska klassisella teorialla ei pystytä tyydyttävästi selittämään käsitteiden sumeutta, sitä ei enää pidetä kuvailevana eikä normatiivisena teoriana ihmisen muistin rakenteesta [Mur02a, MR05].

Tällä hetkellä uskottavimmat ihmisen muistin rakennetta selittävät teoriat perustuvat vahvasti *prototyypiteoriaan*. Rosch kehitti sen havaittuaan prototyypivaikutuksen [MR05]. Teorian pyrkimyksenä on määritellä käsitteet niin, että prototyypivaikutus on niistä ennustettavissa. Hänen alun perin esittämästään prototyypiteoriasta on olemassa useita tulkintoja, joista yleisin on *summaarinen esitys* (engl. summary representation). Siinä tiettyä kategoriaa kuvailevan käsitteen ajatellaan olevan joukko piirteitä, jotka yleensä ovat havaittavissa tähän kategoriaan kuuluvilla yksilöillä. Näistä piirteistä toiset ovat kategoriaan kuulumisen kannalta merkittävämpiä kuin toiset. Piirteen merkittävyys voi määräytyä esimerkiksi *perheyhtäläisyyden* kautta: mitä useammin piirre  $p$  esiintyy kategoriaan  $K$  kuuluvilla yksilöillä ja mitä harvemmin se esiintyy muihin kategorioihin kuuluvilla yksilöillä, sitä merkittävämpi piirre  $p$  on kategoriaan  $K$  kuulumisen kannalta. Summaarisessa esityksessä piirteet voivat myös olla ristiriitaisia. Esimerkiksi kategoriaa koira,

voivat kuvailla samanaikaisesti piirteet ”koirilla on pitkä karva” ja ”koirilla on lyhyt karva” [GS05].

Summaarisessa esityksessä käsitteen muodostaa siis joukko piirteitä, joista toiset ovat kategoriaan kuulumisen kannalta merkittävämpiä kuin toiset. Yksilö kuuluu tiettyyn kategoriaan, jos sillä voidaan havaita olevan merkittävästi tähän kategoriaan kuuluvilla yksilöillä tavallisesti havaittuja piirteitä. Kategoriaan kuuluminen ei siis ole yhdestä piirteestä kiinni toisin kuin klassisessa teoriassa [GS05]. Lisäksi, toisin kuin klassisessa teoriassa, kategoriaan kuulumiseen voi liittyä epävarmuutta. Näin voi olla silloin, kun yksilöllä on suunnilleen yhtä merkittävästi useaan kategoriaan liittyviä piirteitä [GS05].

Konnektionistiset tietomallit ovat tunnetuimpia laskennallisia tietomalleja, joilla voidaan esittää käsitteitä soveltaen prototyypiteoriaan perustuvia teorioita [Way97, MR05]. Ne ovat verkkoja, joiden painotetuilla kaarilla toisiinsa kytkeytyviä solmuja kutsutaan *neuroneiksi*. Ne jaetaan tyypillisesti kolmeen ryhmään seuraavasti: *syöteneuronit* vastaanottavat informaatiota verkon prosessoitavaksi suoraan ympäristöstä, *piiloneuronit* prosessoivat syötteenä saamansa informaation aktivoimalla toisia verkossa olevia neuroneita ja *tulosteneuronit* heijastavat prosessoinnin tuloksen. Informaation prosessointi tapahtuu aktivaatioiden ja neuronien välisten yhteyksien voimakkuuden määräämällä tavalla [Tho91].

Konnektionistisilla malleilla on luontevaa esittää käsitteet joukkona painotettuja piirteitä. Tiettyä piirrettä vastaa tietyn tai tiettyjen syöteneuronien aktivaatio. Aktivoituessaan syöteneuroni lähettää aktivaatioarvonsa niille piiloneuroneille, joihin siitä on yhteys. Tiettyä käsitettä vastaava tulosneuroni aktivoituu, jos yhteyksiin liittyvien painokertoimien ja niitä pitkin saapuvien aktivaatioarvojen tulojen summa ylittää tietyn kynnysarvon. Tulosneuronin aktivoituminen voi vaatia useiden piiloneuronien aktivoitumista. Konnektionistisilla malleilla käsitteet siis muodostetaan funktionaalisesti piirteistä [RN03, Tho91].

Välttämättömien ja riittävien ehtojen määrittäminen formaalisti voi monesti olla ihmiselle vaikeata. Käsitteiden sumea määrittely formaalisti ei ole ihmiselle sen helpompaa. Ihmiset ymmärtävät huonosti todennäköisyyksiä ja ovat huonoja arvioimaan esimerkiksi kategorisointi päätökseen johtaneiden syiden merkittävyksiä

[LS05]. Niinpä konnektionistiset mallit muodostetaan tyypillisesti erilaisten oppimisalgoritmien avulla [Sun01]. Niiden tarkoituksena on virittää neuronien väliset painokertoimet vastaamaan kuvailtavien yksilöiden piirteiden tilastollisia ominaisuuksia [RN03].

Konnektionistiset mallit eivät ole ongelmattomia algoritmisen soveltamisen näkökulmasta. Tarkoituksenmukaisen prosessoinnin tuottavan konnektionistisen mallin luominen on tavallisesti hidasta oppimisalgoritmeillakin. Hitaus johtuu nykyisten menetelmien vaatimasta suuresta opetuskertojen määrästä. Opetusta varten tarvitaan huomattava määrä riittävän laadukasta harjoitusmateriaalia. Tämä voi osoittautua ongelmalliseksi joillakin sovellusalueilla. Lisäksi ihmisen ja toteutettavan ohjelmiston välisen vuorovaikutuksen kannalta ongelmallista on se, että oppimisalgoritmeilla konnektionistiseen malliin muodostettujen kategorioiden kuvailevien käsitteiden määritelmät ovat kätkeytyneinä verkon neuronien välisten yhteyksien painokertoimiin. Niinpä oppimisalgoritmien muodostamille resursseja kuvaileville käsitteille voi monesti olla vaikea löytää selkeää tulkintaa [FGV01, Way97].

## 4 Kuvauslogiikka semanttisen yhteentoimivuuden perustana

Klassisen teorian puutteista huolimatta siihen perustuvia tietämyksen esitystapoja kehitetään ja sovelletaan jatkuvasti. Klassinen teoria on helposti ymmärrettävä ja elegantti teoria, joka tekee sen soveltamisesta helppoa verrattuna muihin käsitteitä selittäviin teorioihin [LM01, Mur02a]. Uusin klassiseen teoriaan perustuva tietämyksen esityskieli on kuvauslogiikka. Se tarjoaa loogisen perustan käsitteiden merkityksen määrittelyyn ja tulkintaan [BCM<sup>+</sup>03b, BS01]. Kuvauslogiikka on osoittautunut hyväksi perustaksi monien tietämyksen hallintaan liittyvien käytännön ongelmien ratkaisemisessa [BCM<sup>+</sup>03b, MHN04, Hor98].

Kuvauslogiikkaa on käytetty menestyksekkäästi esimerkiksi tietokantojen hallinnassa (engl. database management) tietokantakyselyiden optimoimiseen, tietokannan johdonmukaisuuden (engl. consistency) tarkastamiseen ja näkymien (engl. views) luomiseen sekä niiden hierarkiseen kategorisointiin [BCM<sup>+</sup>03b, Bor95]. Tällä hetkellä kuvauslogiikkaa kehitetään erityisesti Semantic Web -kehitysalueella ja hyödynnetään merkittävästi muiden muassa biolääketieteessä geneettisen tiedon mallinnuksessa [BTMS04, WSGA03].

### 4.1 Käsitteiden määrittely

Kuvauslogiikassa käsitteet määritellään käyttäen *primitiivikäsitteitä* (engl. primitive concepts tai atomic concepts), *-primitiivirooleja* (engl. primitive roles tai atomic roles) (ominaisuuksia) ja *käsitekonstruktoreita* (engl. concept constructors). Minimaa-lisessa peruskuvauslogiikassa ( $\mathcal{AL}$ -kielessä) käsitekonstruktoreiksi on määritelty negaatio, leikkaus, arvorajoite ja rajoitettu eksistentiaalinen kvantifointi.  $\mathcal{AL}$ -kielessä käsittemääritelmät noudattavat seuraavia syntaktisia sääntöjä, joissa  $A$  edustaa primitiivikäsitettä, kirjain  $R$  ominaisuutta ja kirjaimet  $C$  ja  $D$  määriteltyjä käsitteitä (engl. defined concept) eli *koosteisia käsitteitä* [BCM<sup>+</sup>03c]:

$C, D \longrightarrow A$		(primitiivikäsite)
$\top$		(kaikki käsite)
$\perp$		(tyhjä käsite)
$\neg A$		(primitiivinegaatio)
$C \sqcap D$		(leikkaus)
$\forall R.C$		(arvorajoite)
$\exists R.\top$		(rajoitettu eksistentiaalinen kvantifiointi).

Määritellyt käsitteet  $C$  ja  $D$  voivat siis olla rekursiivisesti primitiivikäsite, kaikki käsite, tyhjä käsite, primitiivinegaatio, leikkaus, arvorajoite tai rajoitettu eksistentiaalinen kvantifiointi.

Näillä rekursiivisilla säännöillä muodostettujen käsitelmääritelmien tulkinta  $\mathcal{I}$  on formaalisti määritelty. Tulkinnat koostuvat *määrittelyjoukosta*  $\Delta^{\mathcal{I}}$  ja *tulkintafunktios-  
ta*  $\cdot^{\mathcal{I}}$ . Määrittelyjoukko on epätyhjä, kaikkien mahdollisten yksilöiden muodostama joukko. Tulkintafunktio kuvaa jokaisen primitiivikäsitteen  $A$  määrittelyjoukon osajoukoksi:

$$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}.$$

Vastaavalla tavalla tulkintafunktio kuvaa jokaisen ominaisuuden  $R$  määrittelyjoukon yksilöiden välisten kaikkien mahdollisten binaarirelaatioiden osajoukoksi:

$$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}.$$

Käsitekonstruktoilla muodostetut määritelmät tulkitaan induktiivisesti noudattaen seuraavia sääntöjä [BCM<sup>+</sup>03c]:

$$\begin{aligned} \top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\ \perp^{\mathcal{I}} &= \emptyset \\ (\neg A)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} \\ (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\ (\forall R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \forall b.(a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\ (\exists R.\top)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists b.(a, b) \in R^{\mathcal{I}}\}. \end{aligned}$$

Kaikki käsite tulkitaan määrittelyjoukoksi ja tyjä käsite tulkitaan tyhjäksi joukoksi eli joukoksi, johon ei kuulu yhtään yksilöä. Primitiivikäsitteen  $A$  negaatio tulkitaan kaikkien niiden määrittelyjoukon yksilöiden muodostamaksi joukoksi, johon ei kuulu primitiivikäsitteen  $A$  ekstensioon kuuluvat yksilöt. Leikkauksen täsmällisempi tulkinta on

$$(C \sqcap D)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid x \in C^{\mathcal{I}} \text{ ja } x \in D^{\mathcal{I}}\}$$

eli käsitteiden  $C$  ja  $D$  leikkaus tulkitaan joukoksi yksilöitä, jotka kuuluvat sekä käsitteen  $C$  että käsitteen  $D$  ekstensioon.

Ominaisuuden  $R$  ja käsitteen  $C$  muodostama arvorajoite  $(\forall R.C)$  tulkitaan joukoksi yksilöitä, jotka ovat ominaisuuden  $R$  ilmentämässä relaatiossa ainoastaan käsitteen  $C$  ekstensioon kuuluviin yksilöihin (eli arvorajoite  $\forall R.C$  tulkitaan joukoksi yksilöitä, joiden ominaisuuden  $R$  arvona on ainoastaan käsitteen  $C$  ekstensioon kuuluvia yksilöitä). Toisin sanoen jos näiden yksilöiden ominaisuudella  $R$  on arvo, niin se kuuluu käsitteen  $C$  ekstensioon. Niinpä pelkästään arvorajoitteen  $\forall R.C$  perustella ei näillä yksilöillä voida tulkita olevan vähintään yhtä relaatiota  $R$  käsitteen  $C$  ekstensioon kuuluvaan yksilöön [W3Cc]. Sitä vastoin ominaisuuden  $R$  rajoitettu eksistentiaalinen kvantifiointi tulkitaan joukoksi yksilöitä, joiden ominaisuuden  $R$  arvona on vähintään yksi määrittelyjoukkoon kuuluva yksilö.

Oletetaan esimerkiksi **Ihminen** primitiivikäsitteeksi, joka tulkitaan ihmisten muodostamaksi joukoksi, **Naispuolinen** primitiivikäsitteeksi, joka tulkitaan naispuolisten muodostamaksi joukoksi, ja **omaaAviopuolison** ominaisuudeksi, joka tulkitaan kaikkien ”yksilö  $b$  on yksilön  $a$  aviopuoliso” -parien muodostamaksi joukoksi. Näiden oletusten nojalla  $\mathcal{AL}$ -kielellä voidaan esimerkiksi määritellä käsite nainen siten, että se tulkitaan kaikkien naispuolisten ihmisten muodostamaksi joukoksi:

$$\text{Ihminen} \sqcap \text{Naispuolinen}.$$

Jos oletetaan lisäksi **Miespuolinen** primitiivikäsitteeksi, joka tulkitaan miespuolisten muodostamaksi joukoksi, niin käsite mies voidaan määritellä vastaavalla tavalla:

$$\text{Ihminen} \sqcap \text{Miespuolinen}.$$

Minimaalisesta  $\mathcal{AL}$ -kielestä voidaan tehdä ilmaisuvoimaisempi määrittelemällä siihen lisää konstruktoreita. Esimerkiksi jos  $\mathcal{AL}$ -kieleen lisätään yleinen negaatio niin kielestä tulee ilmaisuvoimaisempi. Sen avulla voidaan määritellä täydellinen eksistentiaallinen kvantifiointi ja käsitteiden yhdiste. Kielestä tulee kuitenkin helppokäyttöisempi, jos näille kaikille kuvauksille määritellään omat käsitekonstruktorit. Yleinen negaatio, yhdiste ja täydellinen eksistentiaallinen kvantifiointi esitetään seuraavasti [BCM<sup>+</sup>03b]:

$$\begin{aligned} \neg C \mid & \quad (\text{yleinen negaatio}) \\ C \sqcup D \mid & \quad (\text{yhdiste}) \\ \exists R.C \mid & \quad (\text{täydellinen eksistentiaallinen kvantifiointi}). \end{aligned}$$

Yleinen negaatio ja täydellinen eksistentiaallinen kvantifiointi laajentavat vastaavat rajoittuneemmat mallinnusprimitiivit (negaation ja rajoitetun eksistentiaallisen kvantifioinnin) koskemaan myös koosteisia käsitteitä, joihin merkinnällä  $C$  viitataan.

Käsitteiden yleinen negaatio, yhdiste ja täydellinen eksistentiaallinen kvantifiointi määritellään tulkintafunktion avulla seuraavasti [BCM<sup>+</sup>03c]:

$$\begin{aligned} (\neg C)^{\mathcal{I}} &= A^{\mathcal{I}} \setminus \Delta^{\mathcal{I}} \\ (C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\ (\exists R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists b.(a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}. \end{aligned}$$

Yhdisteen täsmällisempi tulkinta on

$$(C \sqcup D)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid x \in C^{\mathcal{I}} \text{ tai } x \in D^{\mathcal{I}}\}$$

eli käsitteiden  $C$  ja  $D$  yhdiste tulkitaan joukoksi yksilöitä, jotka kuuluvat käsitteen  $C$ , käsitteen  $D$  tai molempien ekstensioon. Ominaisuuden  $R$  ja käsitteen  $C$  muodostama yleinen eksistentiaallinen kvantifiointi ( $\forall R.C$ ) tulkitaan joukoksi yksilöitä, jotka ovat relaatiossa  $R$  vähintään yhteen käsitteen  $C$  ekstensioon kuuluvaan yksilöön.

Käsite mies voidaan esimerkiksi määritellä yleisen negaation avulla siten, että se tulkitaan joukoksi ihmisiä, jotka eivät ole naispuolisia:

$$\text{Ihminen} \sqcap \neg(\text{Ihminen} \sqcap \text{Naispuolinen}).$$



Täydellisen eksistentiaalisen kvantifoinnin avulla esimerkiksi käsite vaimo voidaan määritellä tulkittavaksi joukoksi naisia, joilla on aviopuolisonaan mies:

$$(\text{Ihminen} \sqcap \text{Naispuolinen}) \sqcap \exists \text{omaaAviopuolison.}(\text{Ihminen} \sqcap \text{Miespuolinen}).$$

Vastaavalla tavalla voimme määritellä käsitteen aviomies:

$$(\text{Ihminen} \sqcap \text{Miespuolinen}) \sqcap \exists \text{omaaAviopuolison.}(\text{Ihminen} \sqcap \text{Naispuolinen}).$$

Käsite aviopuoliso voidaan määritellä yhdisteen avulla joukoksi, jonka muodostavat vaimot ja aviomiehet:

$$\begin{aligned} & ((\text{Ihminen} \sqcap \text{Naispuolinen}) \sqcap \exists \text{omaaAviopuolison.}(\text{Ihminen} \sqcap \text{Miespuolinen})) \\ & \sqcup ((\text{Ihminen} \sqcap \text{Miespuolinen}) \sqcap \exists \text{omaaAviopuolison.}(\text{Ihminen} \sqcap \text{Naispuolinen})). \end{aligned}$$

Jos  $\mathcal{AL}$ -kieleen on lisätty negaatio kaikille käsitteille, niin kielen nimeen lisätään  $\mathcal{C}$ -kirjain. Vastaavasti kielen nimeen lisätään kirjain  $\mathcal{U}$ , jos siihen on lisätty yhdiste ja  $\mathcal{E}$ , jos siihen on lisätty täydellinen eksistentiaalinen kvantifointi. Koska yhdiste ja täydellinen eksistentiaalinen kvantifointi eivät kuitenkaan lisää kielen ilmaisuvoimaa, yleisen negaation lisäämisen jälkeen, niin kaikki edellä kuvatut käsitekonstruktorit sisältävään kieleen viitataan tavallisesti nimellä  $\mathcal{ALC}$  [BCM<sup>+</sup>03c].

Kuvauslogiikassa yksilö  $y$  voidaan määritellä kuuluvaksi käsitteen  $C$  kuvailemaan kategoriaan seuraavasti [HPSvH03]:

$$y \in C.$$

Jos yksilö  $y$  määritellään kuuluvaksi käsitteen  $C$  kuvailemaan kategoriaan, niin sillä voidaan tulkita olevan kaikki käsitteen  $C$  määritelmien mukaiset ominaisuudet ja niiden arvot [HPSvH03]:

$$y^{\mathcal{I}} \in C^{\mathcal{I}}.$$

Tämän lisäksi yksilölle voidaan määritellä ominaisuuksia ja niiden arvoja seuraavasti [HPSvH03]:

$$\langle y, y_i \rangle \in R_i.$$

Yksilön  $y$  ominaisuus  $R$  saa arvon  $y_i$ . Tämän tulkintafunktio on

$$\langle y^{\mathcal{I}}, y_i^{\mathcal{I}} \rangle \in R_i^{\mathcal{I}}$$

eli yksilö  $y$  on ominaisuutta  $R$  vastaavassa relaatiossa yksilöön  $y_i$ .

## 4.2 Käsitteiden automaattinen tulkinta

Kuvauslogiikassa määritellyillä käsitteillä on perusominaisuuksia, jotka luetellaan *aksiomissa*. Määritellyt käsitteet ja aksiomat kiinnittävät yhdessä käsitteiden muodostaman rakenteen eli *käsitejärjestelmän*.  $\mathcal{ALC}$ -kielellä määriteltyjen käsitteiden välisiä aksiomia ovat esimerkiksi niin sanotut *terminologiset aksiomat* [BCM<sup>+</sup>03c, FBS02]: *vastaavuus* (engl. equivalence) ja *sisältyminen* (engl. inclusion).

Vastaavuuden yleinen muoto on

$$\mathbf{C} \equiv \mathbf{D}$$

jossa  $C$  ja  $D$  ovat käsitteitä. Tämä aksioma tulkitaan formaalisti seuraavasti

$$(\mathbf{C} \equiv \mathbf{D})^{\mathcal{I}} = (C^{\mathcal{I}} = D^{\mathcal{I}})$$

eli  $C$  ja  $D$  ovat merkitykseltään vastaavia käsitteitä, jos molempien käsitteiden tulkinta on sama. Sisältymisen yleinen muoto on [BCM<sup>+</sup>03c]

$$\mathbf{C} \sqsubseteq \mathbf{D}$$

ja se tulkitaan formaalisti seuraavalla tavalla

$$(\mathbf{C} \sqsubseteq \mathbf{D})^{\mathcal{I}} = (C^{\mathcal{I}} \subseteq D^{\mathcal{I}})$$

eli käsite  $C$  sisältyy käsitteeseen  $D$ , jos käsitteen  $C$  tulkinta on käsitteen  $D$  tulkinnan osajoukko.

Semanttisen yhteentoimivuuden kannalta keskeistä on pystyä löytämään automaattisesti juuri nämä terminologisten aksioomien implisiittiset ilmentymät eli käsitteiden merkitystä ilmentävien määritelmien väliset implisiittiset vastaavuudet ja sisältymiset.  $\mathcal{ALC}$ -kieleen, kuten muihinkin kuvauslogiikoihin liittyvän formaalisti määritellyn tulkintafunktion avulla on mahdollista selvittää algoritmisesti mihin kategoriaan sillä määritellyt käsitteet viittaavat ja millaisia osajoukkosuhteita näiden kategorioiden välillä on. Niinpä tulkintafunktion avulla on mahdollista selvittää automaattisesti käsitteiden välisiä sisältymisiä ja vastaavuuksia [BS01, Hor02, BCM<sup>+</sup>03c].

Jos  $\mathcal{T}$  oletetaan kuvauslogiikalla määriteltyjen käsitteiden muodostamaksi käsitejärjestelmäksi, niin sen sisältämät implisiittiset käsitteiden väliset sisältymiset ja vastaavuudet voidaan päätellä formaalisti seuraavasti [BCM<sup>+</sup>03c]:

**Vastaavuus** Käsitejärjestelmässä  $\mathcal{T}$  Käsitteet  $C$  ja  $D$  vastaavat toisiaan, jos  $C^{\mathcal{I}} = D^{\mathcal{I}}$  käsitejärjestelmän  $\mathcal{T}$  kaikilla tulkinnoilla.

**Sisältyminen** Käsite  $A$  sisältyy käsitteeseen  $C$ , jos  $A^{\mathcal{I}} \subseteq C^{\mathcal{I}}$  käsitejärjestelmän  $\mathcal{T}$  kaikilla tulkinnoilla.

Jos käsitejärjestelmässä  $\mathcal{T}$  käsitteen  $C$  merkitys sisältyy käsitteen  $D$  merkitykseen ja lisäksi käsitteen  $D$  merkitys sisältyy käsitteen  $C$  merkitykseen, niin käsitteet  $C$  ja  $D$  ovat vastaavia. Vastaavuuden päättely voidaan siis pelkistää sisältymisen päättelyyn (engl. reduction to subsumption). Sisältymisen päättely voidaan edelleen pelkistää toteutuvuuden (engl. satisfiability) päättelyyn. Toteutuvuuden formaali päättely määritellään seuraavasti [BCM<sup>+</sup>03c]:

**Toteutuvuus** Käsitejärjestelmässä  $\mathcal{T}$  käsite  $C$  on toteutuva jos käsitejärjestelmälle  $\mathcal{T}$  on olemassa tulkinta, jossa käsitteen  $C$  tulkinta ( $C^{\mathcal{I}}$ ) on epätyhjä.

Käsitteen  $C$  merkitys sisältyy käsitteen  $D$  merkitykseen käsitejärjestelmässä  $\mathcal{T}$ , jos käsitteen  $C$  ja käsitteen  $D$  negaation leikkaus eli  $C \sqcap \neg D$  ei ole toteutuva.

Terminologisten aksiomien ilmentymiä eksplisiittisesti esittämällä voidaan liittää toisiinsa käsite ja sen määritelmä [FBS02]. Voidaan esimerkiksi esittää, että käsitettä vaimo vastaa määritelmä, joka tulkitaan aviopuolisonaan miehen omaavaksi naiseksi:

$$\text{Vaimo} \equiv \text{Nainen} \sqcap \exists \text{omaaAviopuolison.Mies},$$

Aiemmassa esimerkissä käsite **Aviopuoliso** määriteltiin kuvauslogiikalla käsitteiden **Vaimo** ja **Aviomies** yhdisteeksi. Tulkintafunktion avulla käsite **Aviopuoliso** voidaan tulkita kategoriaksi, jonka muodostavat käsitteiden **Vaimo** ja **Aviomies** tulkintoja vastaavat kategoriat. Niinpä esimerkiksi vaimojen muodostama kategoria on aviopuolisojen muodostaman kategorian osajoukko eli voidaan päätellä, että käsitteen **Vaimo** merkitys sisältyy käsitteen **Aviopuoliso** merkitykseen.

Eri toimijoiden on mahdollista tulkita toistensa kuvauslogiikalla määrittelemät käsitteet automaattisesti, jos ne perustavat määrittelynsä mahdollisimman muuttumattomaan yhteiseen primitiivikäsitteiden ja ominaisuuksien muodostamaan joukkoon eli *primitiivikäsitteistöön*. Tällä *kuvauslogiikka+primitiivikäsitteistö-menetelmällä* eri toimijat voivat määritellä toisistaan riippumattomasti kuvauslogiikalla primitiivikäsitteistä ja ominaisuuksista uusia käsitteitä kuvailemaan uusia olioita ja säilyttää semanttisen yhteentoimivuuden. Erityisesti jos käsitteiden määrittelyt ovat tosia ja eheitä niin voidaan taata, että täydellinen (engl. complete) ja pätevä (engl. sound) sisältymispäätely löytää vain ja ainoastaan kaikki todet semanttiset suhteet käsitteiden väliltä [Rec03]. Primitiivikäsitteet ja ominaisuudet ovat siis eräänlaisia uudelleenkäytettäviä käsitteiden rakennuspalikoita.

On kuitenkin tärkeää huomata, että toimijan  $t_1$  määrittelemien käsitteiden tulkinta on tarkoitettu toimijalle  $t_2$  vain silloin, kun toimijalla  $t_2$  on tiedossa toimijan  $t_1$  käsitteiden määrittelyssä käyttämien primitiivikäsitteiden ja ominaisuuksien tarkoitettu tulkinta. Myös ihmisten kommunikointi luonnollisella kielellä perustuu suureksi osaksi riittävän suureen joukkoon yhdisteltäviä käsitteitä ja ominaisuuksia, joille tunnetaan samat merkitykset. Yhdistelemällä tällaisia primitiivikäsitteitä ja ominaisuuksia voidaan luoda uusia käsitteitä, joiden merkityksen muut pystyvät tulkitsemaan. Useimpien käsitteiden merkitystä ei siis tarvitse muistaa, vaan se voidaan päätellä käsitteen muodostavien primitiivikäsitteiden ja ominaisuuksien merkitysten perusteella [MR05].

Jos eri toimijat päätyvät jostakin syystä käyttämään eri primitiivikäsitteistöjä, niin näiden väliset vastaavuudet on ihmisvoimin määriteltävä semanttisen yhteentoimivuuden toteuttamiseksi. Tämä on kuitenkin todennäköisesti kertaluontoisempi ja helpompi prosessi kuin sellaisten käsitteistöjen tapauksessa, joita ei ole määriteltä käyttäen primitiivikäsitteistöä. Primitiivikäsitteet ovat yleisiä ja mahdollisimman muuttumattomia, joten eri primitiivikäsitteistöt vastaavat huomattavasti todennäköisemmin enemmän ja pidempään toisiaan kuin toimialakohtaiset käsitteistöt.

### 4.3 Esimerkki kuvauslogiikan soveltamisesta

Seuraavan esimerkin toimialue on hieman epätavallinen, mutta se toimii hyvänä analogisena esimerkkinä eri toimialojen tyypillisistä ongelmista, jotka on mahdollista ratkaista kuvauslogiikan avulla.

Viljakäärmeet ovat pääasiassa lemmikeiksi myytäviä kuristajakäärmeitä, joiden hinta määräytyy oleellisesti niiden värityksen perusteella. Mitä harvinaisempi väritys käärmeysilöllä on, sitä arvokkaampi se on rahallisesti. Yksinkertaistetusti voidaan sanoa, että viljakäärmeen värityksen synnyttävät valkoiselle pohjavärille mustaa ja punaista pigmenttiä tuottavat geenit. Erilaiset väritykset syntyvät pigmenttejä tuottavien geenien mutaatioiden ja näiden kombinaatioiden aiheuttamien *värimuunnosten* tuloksena.

Viljakäärmeellä on kategoriaan **Anerytristinen** kuuluva värimuunnos, jos käärme on perinyt molemmilta vanhemmiltaan punaisen pigmentin tuottamiseen geenit, jotka eivät mutatoitumisen seurauksena enää toimi. Jos viljakäärme on perinyt molemmilta vanhemmiltaan mutatoitumisen seurauksena mustaa pigmenttiä normaalia vähemmän tuottavat geenit, niin sillä on kategoriaan **Hypomelanistinen** kuuluva värimuunnos. Jos käärme on perinyt vähintään toiselta vanhemmaltaan muuttumattoman punaista ja mustaa pigmenttiä tuottavan geenin, niin käärmeellä ei ole näiden pigmenttien osalta värimuunnosta. Tällöin sanotaan, että käärmeen väritys on punaisen ja mustan pigmentin osalta **Normaali**. Viljakäärmeen väritykseen voivat vaikuttaa myös useat muut mutaatioista johtuvat värimuunnokset, mutta seuraavassa yksinkertaisessa esimerkissä keskitytään tarkastelemaan ainoastaan edellä esitettyjä värimuunnoksia.

Käärmeakauppiaat viittaavat myymiinsä viljakäärmeisiin, joiden väritys on seurausta (ainoastaan) Anerytristisestä värimuunnoksesta muiden muassa käsitteillä Anerythristic, Anery, Anery A, Black Albino, Pastel tai Pastel Pink. Käsitteillä Pastel tai Pastel Pink viitataan myös viljakäärmeisiin, joiden väritys on Anerytristisen ja Hypomelanistisen värimuunnoksen yhteisvaikutuksen tulos. Useimmiten tällaisiin käärmeisiin viitataan kuitenkin käsitteellä Ghost. Tämä aiheuttaa luonnollisesti sekaannusta ja tekee käärmeiden värityksen harvinaisuuden ja edelleen niiden rahallisen arvon arvioimisesta hankalaa tai mahdotonta [VMS].

Jos väritykseltään tietynlaisiin käärmeisiin viittaavat myyntikäsitteet määriteltäisiin kuvauslogiikalla, käyttäen primitiivikäsitteinä käärmeiden värityksen ja toisaalta niiden rahallisen arvon määrääviä värimuunnoksia, niin eri myyjien käyttämät myyntikäsitteet olisi mahdollista suhteuttaa automaattisesti toisiinsa värimuunnosten perusteella. Ostajan olisi esimerkiksi mahdollista löytää automaattisesti samaa tarkoittavat myyntikäsitteet ja myyntikäsitteiden ylä- ja alakäsitteet. Lisäksi ostajien olisi mahdollista etsiä myytäviä käärmeitä tuntemiensa värimuunnosten perusteella merkitykseltään päällekkäisten ja mahdollisesti monimerkityksellisten myyntikäsitteiden sijaan.

Oletetaan omaaVärimuunnoksen ominaisuudeksi, joka tulkitaan ”yksilö  $y$  omaa värimuunnoksen  $v$ ” -parien joukoksi. Jos nyt esimerkiksi toimijan  $t_1$  käyttämän myyntikäsitteen Anery määritelmä on

$$\text{Anery} \equiv \text{Viljakäärme} \sqcap \exists \text{omaaVärimuunnoksen.Anerytristinen}$$

ja toimijan  $t_2$  käyttämän käsitteen Black Albino määritelmän on

$$\text{Black Albino} \equiv \text{Viljakäärme} \sqcap \exists \text{omaaVärimuunnoksen.Anerytristinen},$$

niin käsitteiden merkitys voidaan selvästikin automaattisesti tulkita samaksi.

Jos toimija  $t_1$  myy lisäksi käärmeitä, joihin hän viittaa käsitteellä Ghost ja sen määritelmä on

$$\begin{aligned} \text{Ghost} \equiv \text{Viljakäärme} \sqcap \exists \text{omaaVärimuunnoksen.Anerytristinen} \\ \sqcap \exists \text{omaaVärimuunnoksen.Hypomelanistinen} \end{aligned}$$

niin se voidaan automaattisesti tulkita **Anery** ja **Black Albino** kategorioiden alikategoriaksi. Toisin sanoen voidaan automaattisesti tulkita, että **Ghost** kategoriaan kuuluvilla käärmeillä on **Anery** ja **Black Albino** käärmeillä olevan värimuunnoksen lisäksi toinen värimuunnos. Niinpä voidaan edelleen päätellä toimialaan liittyvien taustaoletusten nojalla, että kategoriaan **Ghost** kuuluvien käärmeiden täytyy olla harvinaisempia ja siten myös todennäköisesti rahallisesti arvokkaampia, kuin sen yläkategoriaan kuuluvien käärmeiden.

## 5 Yläontologia primitiivikäsitteistönä

Toimialariippumattoman semanttisen yhteentoimivuuden toteuttamiseksi primitiivikäsitteistön tulisi muodostaa joukko käsitteitä ja niihin liittyviä ominaisuuksia, joista on mahdollista määritellä käsitteitä kuvailemaan olioita kaikilta toimialoilta riittävän tarkasti. Useimmiten palveluita ja muita olioita kuvailevat riittävän tarkasti seuraavaksi tarkasteltavat *perustason käsitteet*.

### 5.1 Perustason käsitteet

Oletetaan hierarkkinen käsitteistö  $K$ , joka sisältää johonkin toimialaan läheisesti liittyviä olioita kuvailevia käsitteitä. Hierarkian ylimpänä on kaikkia olioita kuvaileva käsite. Tälle käsitteelle on määritelty asteittain erityisempiä alakäsitteitä kuvailemaan asteittain pienempiä, piirteiltään samankaltaisten olioiden muodostamia joukkoja. Käsitteistön  $K$  yleisimmästä käsitteistä erityisimpiin, alakäsitesuhteita pitkin, kulkevien polkujen likimain keskimmäiset käsitteet ovat perustason käsitteitä. Ne ovat käsitteitä, joita keskivertoihmiset pääosin käyttävät kommunikoidessaan keskenään [Mur02c].

Erityisesti asiaa tai esinettä kuvailevista käsitteistä on selvästi havaittavissa perustason käsitteet. Jos ihmiselle näytetään valokuvia erilaisista olioista ja häntä pyydetään kuvailemaan näkemänsä, niin on erittäin todennäköistä, että hän kuvailee näkemänsä oliot perustason käsitteillä. Esimerkiksi jos ihmiselle näytetään valokuvaa cornish rexistä, niin hän todennäköisimmin kuvailee näkemänsä käsitteellä kissa, joka on perustason käsite. Vain harva ihminen kuvailisi näkemänsä käsitteellä cornish rex, joka on käsitteen kissa alakäsite. Vielä harvempi ihminen kuvailisi näkemänsä, käsitteen kissa yläkäsitteellä, eli esimerkiksi käsitteellä eläin [Mur02c].

Vastaavasti jos ihminen tarvitsee tietokoneen korjaus -palvelua, niin on epätodennäköistä, että hän kuvailisi tämän tarpeensa tyydyttävän palvelun yleisellä käsitteellä ”korjaaminen”. Vielä epätodennäköisempää on, että hän kuvailisi sen hyvin erityisellä käsitteellä ”BIOS-muistipiirin uudelleenohjelmointi” tai ”Millenia S300n -pöytä tietokoneen korjaaminen”. Todennäköinen käsite tässä tapauksessa kuvailuun olisi perustason käsite ”tietokoneen korjaus”.



Fyysiset kohteet toimivat usein tarkentimina prosesseja kuvailevissa käsitteissä, kuten edellä esitetyissä palvelukäsitteissä. Esimerkiksi käsite ”tietokoneen korjaaminen”, koostuu melko yleisestä käsitteestä ”korjaaminen” ja tarkenteesta ”tietokoneen”, joka on perustason käsite ja määrittelee korjaamisprosessin kohteen. Kokonaisuutena palvelukäsite ”tietokoneen korjaaminen” on perustason käsite. Jos yleiseen prosessikäsitteeseen on liitetty useampia perustason käsitteitä tarkentimiksi, niin kokonaisuutena käsite on perustason käsitettä erityisempi. Niinpä primitiivikäsitteistöllä, jolla on mahdollista määritellä perustason käsitteitä yleisesti on mahdollista määritellä runsaasti hyvin erityisiä käsitteitä.

Ihmiset käyttävät kommunikoidessaan pääosin perustason käsitteitä, koska ne ovat eri käsitetasojen käsitteistä erilaistuneimpia (engl. most differentiated). Perustason käsitteisiin liittyy paljon kategorialla kuvailevaa informaatiota. Niitä yleisempiin käsitteisiin liittyy huomattavasti vähemmän kategorialla kuvailevaa informaatiota. Perustason käsitteitä alemman tason käsitteet ovat erityisempiä ja siten niihin liittyy enemmän kategorialla kuvailevaa informaatiota kuin perustason käsitteisiin. Perustason käsitteitä alemman tason käsitteet lisäävät perustason käsitteisiin kuitenkin vain tietoa olion yksityiskohdista, joten kunkin perustason alakäsitteet eroavat vain vähän toisistaan eli ne eivät ole erityisen erottelevia (engl. distinctive). Perustason käsitteet taas sisältävät paljon informaatiota verrattuna niitä ylemmän tason käsitteisiin ja ovat siten erottelevia. Perustason käsitteitä ylemmän tason käsitteet ovat vielä perustason käsitteitäkin erottelevampia, mutta ne eivät ole informatiivisia. Ainoastaan perustason käsitteet ovat sekä informatiivisia että erottelevia eli ne ovat eri käsitetasojen käsitteistä erilaistuneimpia [Mur02c].

Perustason käsitteiden erilaistuneisuus tekee niistä tehokkaita välineitä kommunikointiin [Mur02c]. Niinpä niiden merkitysten automaattisella tunnistamisella on keskeinen asema semanttisen yhteentoimivuuden toteuttamisessa.

## 5.2 Primitiivikäsitteet

Primitiivikäsitteistön tulisi muodostaa joukko käsitteitä ja niihin liittyviä ominaisuuksia, joista on mahdollista määritellä vähintään perustason käsitteitä kuvailemaan olioita kaikilta toimialoilta. Primitiivikäsitteiden muodostaman joukon tulisi

siis koostua perustason käsitteitä yleisemmistä käsitteistä. Primitiivikäsitteistössä ei saisi esiintyä käsitteitä, jotka ovat selvästi muiden käsitteistön käsitteiden avulla määriteltävissä. Toisin sanoen yleisen semanttisen yhteentoimivuuden toteuttamisessa käytettävän primitiivikäsitteistön tulisi muodostaa mahdollisimman toisteeton joukko yleisiä käsitteitä, joilla on mahdollista määritellä perustason käsitteitä kuvailemaan mihin tahansa toimialaan liittyviä olioita.

Yleiset käsitteet kuvailevat tavallisesti ei-laskettavia asioita tai esineitä. Esimerkiksi kaluste, vaatetus ja ruoka ovat yleisiä käsitteitä. Laskettavia asioita tai esineitä kuvailevat käsitteet, esimerkiksi tuoli, pöytä, paita, housut ja omena ovat taas tyyppillisesti perustason käsitteitä tai niiden alakäsitteitä. Tämä on havaittavissa likimain kaikissa kielissä. Ihmiset tulkitsevat perustason käsitteet siten, että ne kuvailevat yksilöä eli yksittäistä olioita. Yleiset käsitteet taas tulkitaan siten, että ne kuvailevat kategorioita [Mur02c]. Yleiset käsitteet soveltuvat siis hyvin primitiivikäsitteiksi. Niitä erikoistamalla voidaan muodostaa perustason käsitteitä, joilla on luontevaa kuvailla yksittäisiä olioita.

Jos kahden käsitteen määritelmässä esiintyy sama primitiivikäsite, niin niiden välillä on semanttinen yhteys. Mitä yleisemmistä käsitteistä perustason käsitteet määritellään, sitä enemmän semanttisia suhteita määritelyjen käsitteiden välille syntyy. Tämä seuraa siitä, että mitä yleisempiä käsitteiden määrittelyyn käytettävät primitiivikäsitteet ovat, sitä vähemmän niitä toisteettomuuden vaatimuksesta johtuen on ja sitä todennäköisemmin tietty määrittelyyn käytettävä primitiivikäsite esiintyy käsitteen määritelmässä. Toisaalta mitä yleisempiä käsitteiden määrittelyyn käytettävät primitiivikäsitteet ovat sitä monimutkaisempia käsitteiden määritelmistä tulee.

Edellä kuvattua voidaan havainnollistaa seuraavalla analogisella esimerkillä. Luonnollisen kielen sanoja voidaan esittää muiden muassa aakkoskirjoituksella, tavukirjoituksella ja sanakirjoituksella. Kuvitellaan nyt, että suomen kielen sanat vastaavat käsitteitä, sanojen esittäminen vastaa niiden semantiikan määrittelyä ja aakkosto, tavusto ja sanasto vastaavat vaihtoehtoisia primitiivikäsitteiden joukkoja, joiden alkioilla sanoja voidaan määritellä. Aakkostossa siis yksi aakkonen, tavustossa yksi tavu ja sanastaossa yksi sana vastaa yhtä primitiivikäsitettä. Niinpä tässä kuvitteellisessa tilanteessa aakkosto vastaa melko yleistä primitiivikäsitteistöä, tavusto

vastaa vähemmän yleistä primitiivikäsitteistöä ja sanasto vastaa äärimmäisen ei-yleistä primitiivikäsitteistöä.

Jos käsitteet määritellään aakkosilla, niin käsitteiden välille syntyy runsaasti semanttisia yhteyksiä, koska aakkosia on suomen kielen sanojen määrään nähden erittäin vähän ja siten sanoissa esiintyy useasti samoja aakkosia. Jos käsitteitä määritellään tavustolla, niin käsitteiden välille syntyy semanttisia yhteyksiä, mutta huomattavasti vähemmän kuin aakkostolla määriteltäessä. Tämä johtuu siitä, että tavuja on suomen kielen sanojen määrään nähden merkittävästi enemmän kuin aakkosia. Jos käsitteitä määritellään sanastolla, niin käsitteiden välille ei synny ollenkaan semanttisia suhteita, koska yksi sana määrittelee yhden käsitteen, jolloin kahdessa eri sanassa ei voi koskaan esiintyä samaa sanaa.

Primitiivikäsitteiden yleisyyden asteen kasvaessa korkeaksi perustason käsitteiden määrittämisestä voi tulla hyvin työlästä tai mahdotonta. Primitiivikäsitteiden käytännön soveltamisen kannalta on erityisen tärkeää löytää niille sellainen yleisyyden aste, jolla käsitteiden määrittely ei muodostu käytännössä ylivoimaiseksi ja jolla määrittelyiden myötä syntyy myös eri toimialojen käsitteiden välille semanttisia yhteyksiä.

### 5.3 Yläontologiat primitiivikäsitteistönä

Erikoistettavaksi tarkoitettuja yleisiä käsitteitä on systemaattisesti määritelty niin sanotuissa *yläontologioissa* (engl. upper ontology, upper level ontology, top ontology). Tietojenkäsittelytieteessä ontologialla tarkoitetaan ”käsitteistön eksplisiittistä määrittelyä” [Gru93]. Yläontologioissa on määriteltynä nimenomaan yleisiä toimialariippumattomia käsitteitä, jotka on tarkoitettu toimialakohtaisten käsitteistöjen perustaksi tai avuksi yhdistettäessä kahta tai useampaa toimialakohtaista käsitteistöä [NT03]. Yläontologioita voidaan siis käyttää pohjana, määriteltäessä käsitteistöjä kuvailemaan johonkin toimialaan liittyviä olioita.

Tunnetuimpia yläontologioita ovat OPENCYC<sup>7</sup>, a Descriptive Ontology for Linquis-

<sup>7</sup><http://www.opencyc.org/release/doc/>

tic and Cognitive Engineering (DOLCE)<sup>8</sup>, ja Suggested Upper Merged Ontology<sup>9</sup> (SUMO) [OPB05]. OPENCYC-ontologia on alun perin kehitetty tietämuskannaksi NLP-menetelmien tueksi. Se sisältää noin 5000 käsitettä ja on näistä ontologioista suurin. OPENCYC-ontologia sisältää runsaasti jopa perustason käsitteitä erityisempiä käsitteitä [OAH<sup>+</sup>06, OPB05]. Runsaasta perustason ja niitä erityisempien käsitteiden määrästä seuraa runsas käsitteellinen toisteisuus eli OPENCYC-ontologia sisältää runsaasti käsitteitä, jotka olisi selvästi muiden sen sisältämien käsitteiden avulla määriteltävissä. Niinpä OPENCYC-ontologia ei tarjoa selkeää käsitteiden kokonaisuutta koosteisten käsitteiden määrittelyyn. Runsa käsitteellinen toisteisuus tekevät tekevät OPENCYC-ontologiasta huonosti primitiivikäsitteistöksi soveltuvan ontologian.

DOLCE-ontologian kehittämisen tavoitteena on ollut esittää täsmällisesti luonnolliseen kieleen sisältyvät ontologiset käsitteet [OPB05]. DOLCE sisältää kuitenkin ainoastaan noin 40 käsitettä ja siten sen erityisemmätkin lehtitason käsitteet ovat hyvin yleisiä (engl. abstract) [OPB05, OAH<sup>+</sup>06]. Niinpä perustason käsitteiden täsmällinen määrittely välttämättömistä ja riittävästä kategoriaan kuulumisen ehdoista DOLCE-ontologian avulla on erittäin monimutkaista tai mahdotonta.

SUMO-ontologia tarjoaa DOLCE-ontologiaa huomattavasti kattavamman ja yksityiskohtaisemman käsitteistön. SUMO-ontologia on muodostettu yhdistämällä muiden muassa seuraavat ontologiat: [OAH<sup>+</sup>06]: John Sowan yläontologia [Sow00], Russell ja Norvigin yläontologia [RN95], James Allenin aika-aksiomat (engl. temporal axioms) [All84], Casati ja Varzin aukkojen formaali teoria (engl. formal theory of holes) [CV95] ja Barry Smithin raja-aitojen ontologia (engl. ontology of boundaries) [Smi96], Nicola Guarinon formaali mereologia (engl. formal mereology) [BGM96]. Nämä ontologiat on yhdistetty yhdeksi hierarkiaksi, joka sisältää noin 1000 käsitettä.

SUMO-ontologian yksi merkittävimmistä vahvuuksista primitiivikäsitteistönä on sen sopiva käsitteiden erikoistuneisuuden taso. Sen kategorioita tarkimmin kuvailevat käsitteet eivät ole liian erikoistuneita kuten OPENCYC-ontologiassa, eivätkä ne toisaalta ole myöskään liian yleisiä, kuten DOLCE-ontologiassa. SUMO-ontologian hie-

---

<sup>8</sup><http://www.loa-cnr.it/DOLCE.html>

<sup>9</sup><http://ontology.teknowledge.com/>

rarkian lehtitason käsitteet ovat sopivasti hieman perustason käsitteitä yleisempiä. Niinpä sen lehtitason käsitteistä on luontevaa määritellä kuvauslogiikalla perustason käsitteitä eli muodostaa kuvauslogiikalla lehtitason käsitteiden ja niille määritelyjen ominaisuuksille avulla käsitteen määritteleviä kategoriaan kuulumisen riittäviä ja välttämättömiä ehtoja. Toinen SUMO-ontologian vahvuuksista on se, että sen sisältämille käsitteille on määritelty monipuolisesti ominaisuuksia, jotka ovat käsitteiden kuvauslogiikkaperustaisen määrittelyn kannalta välttämättömiä.

## 6 Kuvauslogiikka käytännön web-sovelluksissa

Primitiivikäsitteistö yhdessä kuvauslogiikan kanssa tarjoaa perustan tietoteknisten sovellusten semanttiselle yhteentoimivuudelle. Web ja XML-esitysmuoto tarjoaa perustan toimijoiden keskinäiselle tiedon vaihdolle eli syntaktiselle yhteentoimivuudelle. Webiin soveltuvia tietämyksen esityskieliä kehitetään tällä hetkellä erityisesti Semantic Web -kehitysalueella. Tässä keskitymme tarkastelemaan Web Ontology Language<sup>10</sup> (OWL) -nimisen kielen OWL DL (OWL-kuvauslogiikka) -nimistä muotoa.

OWL-kielen kaksi muuta muotoa ovat OWL Lite ja OWL Full. Ne ovat kaikki XML-esitysmuotoon sarjallistuvia kuvauslogiikoita. OWL-kuvauslogiikassa on pyritty löytämään käytännön sovellusten kannalta sopiva tasapaino määrittelyiden syntaktisen helppouden, ilmaisuvoiman ja laskennallisen monimutkaisuuden välillä [HPSvH03].

Ilmaisuvoimaltaan OWL-kuvauslogiikka sijoittuu OWL Lite ja OWL Full -kielen väliin siten, että OWL Lite on ilmaisuvoimaltaan heikoin ja OWL Full vahvin. Vaikka OWL Lite -kielen ilmaisuvoima on vain hieman OWL-kuvauslogiikan ilmaisuvoimaa heikompi, niin siinä määritellyt konstruktorit ovat OWL-kuvauslogiikassa määritellyjä konstruktoreita huomattavasti rajoittuneemmat. Näistä OWL Lite -kielen syntaktisista rajoitteista johtuen tietämyksen kuvailu sillä on joissakin tapauksissa huomattavasti monimutkaisempaa kuin OWL-kuvauslogiikalla [HPSvH03, dBLPF05].

Toisin kuin OWL-kuvauslogiikalla esitetystä tietämyksestä OWL Full -kielellä esitetystä tietämyksestä on mahdollista esiintyä päättelyongelmia, jotka ovat ratkeamattomia<sup>11</sup> (engl. undecidable). Tämä voi olla ongelmallista joidenkin käytännön sovellusten kannalta [HPSvH03].

---

<sup>10</sup><http://www.w3.org/2004/OWL/>

<sup>11</sup>Ongelma on ratkeamaton, jos se on täydellinen ja pätevä, mutta sille ei ole olemassa äärellisessä ajassa päättyvää päättelyprosessia [MHN04].

## 6.1 OWL-kuvauslogiikka

Resource Description Framework<sup>12</sup> (RDF) on tietomalli, jolla on mahdollista määrittellä subjekti-predikaatti-objekti-kolmikoita eli semanttisia verkkoja webiin. Kolmikoiden määrittely tapahtuu resurssitunnisteiden (engl. Universal Resource Identifiers) (URI) ja literaalien avulla [W3Cd]. URI-tunnisteet ovat tunnisteita, joiden avulla webissä voidaan viitata esimerkiksi web-sivuihin, sähköpostilaatikoihin ja sähköisiin dokumentteihin. Viitattavia resursseja voivat olla myös ihmiset, yritykset ja muut oliot [W3Ce, W3Ca], joten URI-tunnisteet voivat toimia termien tavoin viitauksina mihin tahansa olioihin.

Yhteiskäyttöisen tietämyksen esittämiseen RDF-tietomalli tarjoaa vain vähän semantiikaltaan hyvin määriteltyjä mallinnusprimitiivejä [Hor02]. OWL-kuvauslogiikka sisältää käsitteiden määrittelyyn tarkoitettuja formaalisti tulkittavia käsitekonstruktoireita, ominaisuuskonstruktoireita ja aksioomia, joihin voidaan viitata URI-tunnisteilla. Näiden OWL-kuvauslogiikassa määriteltyjen URI-tunnisteilla viitattavia mallinnusprimitiivien ja RDF-tietomallin avulla voidaan muodostaa kuvauslogiikan lauseita webiin.

Monia käytännön sovelluksia varten  $\mathcal{ALC}$ -kuvauslogiikka ei ole riittävän ilmaisuvoimainen. Se ei esimerkiksi sisällä mallinnusprimitiivejä ja tulkintafunktioita *tietotyyppien* (engl. datatype) ja *kardinaliteettien* (engl. cardinality) määrittelyyn ja tulkintaan, mikä on selkeä puute monien käytännön sovellusten kannalta [HPSvH03, MHN04, BCM<sup>+</sup>03c]. OWL-kuvauslogiikka vastaakin ilmaisuvoimaltaan  $\mathcal{ALC}$ -kuvauslogiikka ilmaisuvoimaisempaa  $\mathcal{ALCR}^+\mathcal{HOIN}(\mathcal{D})$ -kuvauslogiikkaa, joka sisältää muiden muassa konstruktorit ja tulkintafunktiot sekä tietotyyppien, että kardinaliteettien määrittelyyn ja tulkintaan [Hor02, HPSvH03].

$\mathcal{ALCR}^+\mathcal{HOIN}(\mathcal{D})$ -kuvauslogiikassa tietotyypit tulkitaan kaikkien mahdollisten tietoarvojen (engl. data value) muodostaman määrittelyjoukon ( $\Delta_D$ ) osajoukoiksi

$$D^{\mathcal{I}} \subseteq \Delta_D^{\mathcal{I}}$$

jossa  $D$  on tietotyyppi. Esimerkiksi tietotyyppi ”kokonaisluku” tulkitaan kaikkien kokonaislukujen muodostamaksi määrittelyjoukon osajoukoksi, jonka yksi jäsen on

---

<sup>12</sup><http://www.w3.org/RDF/>

esimerkiksi tietoarvo 35. Kaikkien mahdollisten tietotyyppien muodostama määrittelyjoukko on kaikkien mahdollisten yksilöiden muodostamasta määrittelyjoukosta erillinen joukko. Niinpä tietotyyppien automaattinen tulkinta toteutetaan tavallisesti muusta päättelystä erillisenä palveluna. Tämän palvelun tulee sisältää tietotyyppijä koskeva tietämys, jonka perusteella se pystyy antamaan totuusarvoisia vastauksia muun päättelyn mahdollisesti esittämiin tietotyyppijä koskeviin kysymyksiin. Kysymykset ovat tavallisesti muotoa: kuuluuko tietoarvo  $t$  tietotyyppin  $D$  määrittelemään määrittelyjoukon osajoukkoon [HPSvH03].

Kardinaliteetit jakautuu *minimikardinaliteetteihin* ja *maximikardinaliteetteihin*, jotka esitetään seuraavasti

$$\begin{aligned} &\geq nR \mid && \text{(minimikardinaliteetti)} \\ &\leq nR \mid && \text{(maximikardinaliteetti)}. \end{aligned}$$

Niiden tulkintafunktiot ovat [HPSvH03]

$$\begin{aligned} (\geq nR)^{\mathcal{I}} &= \{x \mid \#(\{y. \langle x, y \rangle \in R^{\mathcal{I}}\}) \geq n\} \\ (\leq nR)^{\mathcal{I}} &= \{x \mid \#(\{y. \langle x, y \rangle \in R^{\mathcal{I}}\}) \leq n\} \end{aligned}$$

eli ominaisuuden  $R$  ja numeron  $n$  muodostama numerorajoite ( $\geq nR$ ) tulkitaan joukoksi yksilöitä, jotka ovat ominaisuutta  $R$  ilmentävässä relaatiossa lukumäärältään suurempaan (tai pienempään) tai yhtä suureen määrään yksilöiden tai tietoarvojen määrittelyjoukon ( $\Delta$  tai  $\Delta_D$ ) jäseniä.

Tietotyyppien lisäksi  $\mathcal{ALCR}^+\mathcal{HOIN}(\mathcal{D})$ -kuvauslogiikka tarjoaa mahdollisuuden määrittellä käsitteitä luettelemalla kaikki sen ekstensioon kuuluvat yksilöt. Näin määritellyjä käsitteitä kutsutaan luetelluiksi käsitteiksi (tai luokiksi) (engl. Enumerated Classes) [W3Cb].  $\mathcal{ALCR}^+\mathcal{HOIN}(\mathcal{D})$ -kuvauslogiikassa on myös mahdollista määrittellä ominaisuuden mahdollisten arvojen joukko vastaavalla tavalla, luettelemalla siihen kuuluvat yksilöt. Lueteltu käsite ja ominaisuuden mahdollisten arvojen joukko esitetään seuraavasti:

$$\begin{aligned} A = \{x_1, \dots, x_n\} \mid & \text{(lueteltu käsite)} \\ \{x_1, \dots\} \mid & \text{(ominaisuuden mahdollisten arvojen määrittely)}. \end{aligned}$$

Niiden tulkintafunktiot ovat [HPSvH03]



$$\begin{aligned} A^{\mathcal{I}} &= \{x_1^{\mathcal{I}}, \dots, x_n^{\mathcal{I}}\} \\ \{x_1, \dots\}^{\mathcal{I}} &= \{x_1^{\mathcal{I}}, \dots\}. \end{aligned}$$

eli käsite  $A$  tulkitaan yksilöiden  $x_1, \dots$  ja  $x_n$  muodostamaksi joukoksi. Vastaavalla tavalla mahdollisten arvojen joukko tulkitaan lueteltujen yksilöiden muodostamaksi joukoksi.

$\mathcal{ALCR}^+ \mathcal{HOIN}(\mathcal{D})$ -kuvauslogiikassa on mahdollista esiintyä ja eksplisiittisesti esitellä  $\mathcal{ALC}$ -kuvauslogiikkaan nähden uusia aksioomia. Siinä on esimerkiksi mahdollista määritellä ominaisuushierarkioita, käänteisominaisuuksia ja transitiiivisiä ominaisuuksia. Ominaisuushierarkioiden määrittelyllä tarkoitetaan mahdollisuutta määritellä ominaisuuksien välisiä vastaavuus- ja sisältymissuhteita [dBLPF05].

Käänteisominaisuus ja ominaisuuden transitiiivisuus esitetään seuraavasti [HPSvH03]

$$\begin{aligned} R \equiv (-R_0) \mid & \quad (\text{käänteisominaisuus}) \\ Tr(R) \mid & \quad (\text{transitiiivisuus}) \end{aligned}$$

Jos ominaisuutta  $-R$  vastaava relaatio määritellään käänteisominaisuudeksi ominaisuutta  $R$  vastaavalle relaatiolle ( $R \equiv (-R_0)$ ) määrittelyjoukossa  $\Delta^{\mathcal{I}}$ , niin tällöin seuraava päättely on pätevä

$$(x, y) \in R^{\mathcal{I}} \Leftrightarrow (y, x) \in (-R)^{\mathcal{I}}, \quad \forall x, y \in \Delta^{\mathcal{I}}$$

eli jos ominaisuus  $-R$  on ominaisuuden  $R$  käänteisominaisuus, niin ominaisuutta  $R$  vastaava relaatio yksilöstä  $x$  yksilöön  $y$  vastaa semanttisesti ominaisuutta  $-R$  vastaavaa relaatiota yksilöstä  $y$  yksilöön  $x$ . Esimerkiksi jos ihminen  $x$  määritellään ihmisen  $y$  vaimoksi, niin tästä voidaan päätellä, että ihminen  $y$  on ihmisen  $x$  aviomies ja päin vastoin, koska ominaisuus aviomies on ominaisuuden vaimo käänteisominaisuus.

Jos ominaisuus  $R$  määritellään transitiiiviseksi ( $Tr(R)$ ) määrittelyjoukossa  $\Delta^{\mathcal{I}}$ , niin tällöin seuraava päättely on pätevä

$$(x, y) \in R^{\mathcal{I}} \text{ ja } (y, z) \in R^{\mathcal{I}} \Rightarrow (x, z) \in R^{\mathcal{I}}, \quad \forall x, y, z \in \Delta^{\mathcal{I}}$$

eli jos yksilöstä  $x$  on transitiivista ominaisuutta  $R$  vastaava relaatio yksilöön  $y$  ja yksilöstä  $y$  on ominaisuutta  $R$  vastaava relaatio yksilöön  $z$ , niin tästä seuraa, että yksilöstä  $x$  on ominaisuutta  $R$  vastaava relaatio yksilöön  $z$ .

$ALCR^+$  lyhennetään monesti pelkäksi  $\mathcal{S}$ -kirjaimeksi. Niinpä  $ALCR^+HOIND$ -kuvauslogiikasta käytetään tavallisesti nimeä  $\mathcal{SHOIN}(\mathcal{D})$ . Lyhentämättömässä nimes-  
sä merkintä  $R^+$  viittaa transitiivisiin ominaisuuksiin. Kirjain  $\mathcal{H}$  viittaa ominaisuus-  
hierarkioihin,  $\mathcal{I}$  viittaa käänteisominaisuuksiin,  $\mathcal{N}$  viittaa numerorajoitteisiin ja  $\mathcal{D}$   
viittaa tietotyyppeihin. Kirjain  $\mathcal{O}$  viittaa lueteltuihin käsitteisiin [HE04].

Kielen ilmaisuvoima määräytyy sen sisältämien konstruktoreiden ja sillä esitetys-  
sä tietämyksessä mahdollisesti esiintyvien aksioomien kautta. Kielen ilmaisuvoiman  
lisääntyessä sillä on mahdollista määritellä entistä monimutkaisempaa tietämystä,  
jossa on vastaavasti mahdollista esiintyä entistä monimutkaisempaa implisiittistä  
tietämystä. Tietämyksen monimutkaisuuden lisääntyessä myös implisiittisen tietä-  
myksen päättelyn laskennallinen monimutkaisuus lisääntyy. Tästä seuraa se, että  
päättelyyn vaadittava aika pitenee. Pahimmassa tapauksessa vaadittava aika pite-  
nee äärettömäksi eli päättelyongelmasta tulee ratkeamaton<sup>13</sup> (engl. undecidable).

OWL-kuvauslogiikan kehittelyssä eräs keskeinen tavoite olikin saavuttaa käytännön  
sovellusten kannalta sopiva tasapaino ilmaisuvoiman ja laskennallisen monimutkai-  
suuden välillä. Toisin sanoen OWL-kuvauslogiikan konstruktorit ja aksioomat py-  
rittiin valitsemaan siten, että niillä olisi mahdollista kuvailla riittävän tarkasti käy-  
tännön sovelluksissa tarvittavia käsitteitä ja että implisiittisen tietämyksen päättely  
niillä esitetystä tietämyksestä olisi käytännön sovelluksissa luotettavaa ja riittävän  
nopeaa. Kehittelyn tuloksena syntyi kuvauslogiikka, jolla esitetystä tietämyksessä  
mahdollisesti esiintyvät päättelyongelmat ovat ratkeavia [HPSvH03].

## 6.2 Päättelykoneita OWL-kuvauslogiikalle

Toimijoiden web-perusteisten tietojärjestelmien välisen semanttisen yhteentoimi-  
vuuden toteuttaminen kuvauslogiikka+primitiivikäsitteistö-menetelmällä vaatii web-  
kuvauslogiikan ja primitiivikäsitteistön lisäksi päättelykonetta tunnistamaan ku-

---

<sup>13</sup>Ongelma on ratkeamaton, jos se on täydellinen ja pätevä, mutta sille ei ole olemassa äärellisessä  
ajassa päättyvää päättelyprosessia [MHN04].

vauslogiikalla määriteltyjen käsitteiden välisiä implisiittisiä suhteita automaattisesti. Toteutuvuuden ja sisällymisen päättely ovat standardipalveluita (luokittelupalveluita), joita nykyiset päättelykoneet tarjoavat. Tällä hetkellä käytetyimmät päättelykoneet ovat FaCT++, KAON2, Pellet ja RacerPro [GHT06]. Näistä RacerPro on kaupallinen sovellus ja muut ovat ilmaisia avoimen lähdekoodin sovelluksia [Pan05]. Edellä luetelluista päättelykoneista RacerPro ja FaCT++ ovat keskimäärin tehokkaimpia luokittelijoita [SPG<sup>+</sup>05]. Ne suoriutuvat erilaisista luokittelutehtävistä keskimäärin nopeimmin, FaCT++ kaikista nopeimmin [SPG<sup>+</sup>05]. RacerPro on toteutettu Lisp-ohjelmointikielellä ja FaCT++ on toteutettu C++-ohjelmointikielellä. FaCT++-päättelijän merkittävin heikkous web-sovellusten näkökulmasta on se, että se ei tue OWL-kuvauskielen XML-esitysmuotoa. Muutenkin FaCT++ päättelijän tarjoama rajapinta sen palveluihin on varsin rajoittunut verrattuna muiden tässä mainittujen päättelijöiden rajapintoihin [Pan05].

Pellet on tässä luetelluista päättelijöistä joustavin ja sinnikkäin. Luokiteltaessa se ajautuu virhetilanteeseen tai luovuttaa harvemmin, kuin muut tässä mainitut päättelijät [GHT06, Pan05]. Pellet on keskimäärin RacerPro ja FaCT++-päättelijää hitaampi luokittelija. Se on kuitenkin huomattavasti nopeampi kuin esimerkiksi FaCT++ luokiteltaessa käsitteistöjä, jotka sisältävät runsaasti monimutkaisia käsittemäärittelyjä [SPG<sup>+</sup>05, Pan05]. KAON2 on luokittelijana tehottomin tässä esitellyistä päättelijöistä [GHT06]. Pellet ja KAON2 on toteutettu Java-ohjelmointikielellä [GHT06].

Pellet tarjoaa monipuolisen rajapinnan sen tarjoamien palveluiden hyödyntämiseen. Se tukee XML-esitysmuotoon sarjallistettua kuvauslogiikkaa ja soveltuu siten hyvin käytettäväksi nykyisten web-sovellusten yhteydessä. Lisäksi tässä käsitellyistä päättelykoneista ainoastaan Pellet tukee ilmaisuvoimaltaan täysin *SHOIN(D)*-kuvauslogiikkaa vastaavia kieliä [GHT06, Pan05, SPG<sup>+</sup>05].

## 7 Kuvauslogiikka+primitiivikäsitteistö-menetelmän soveltaminen

Seuraavaksi tarkastellaan ja testataan esimerkkien avulla yhteiskäyttöisten käsitteiden määrittelyä tässä kirjoitelmassa esitetyllä kuvauslogiikka+primitiivikäsitteistö-menetelmällä. Esimerkeissä käytettävät toimialakohtaiset käsitteet ovat peräisin Sosiaali- ja terveysalan tutkimus- ja kehittämiskeskuksen (STAKES) luokituskeskuksen tuottamasta sosiaali- ja terveydenhuollon palveluluokituksen<sup>14</sup> (STAKES-palveluluokitus) versiosta 2.0. Se luokittelee ja kuvailee sosiaali- ja terveydenhuollon toimialan palveluita. Ennen yhteiskäyttöisten käsitteiden määrittelyn tarkastelua perehdytään hieman tarkemmin käsitteeseen palvelu ja STAKES-palveluluokitukseen.

### 7.1 Käsitteen palvelu merkitys

Kehittyneissä teollisuusmaissa toimijoiden on mahdollista ja kannattavaa tyydyttää tarpeitaan kuluttamalla toisten toimijoiden tarjoamia palveluita. *Tarve* on toimijan kokemus siitä, että se on nykytilassaan vailla jotakin aineellista tai aineetonta, tai sillä on tarve päästä eroon jostakin aineellisesta tai aineettomasta. Tyypillisesti agentti pyrkii tarvetta kokiessaan muuttamaan nykytilaansa tavoitteenaan hävittää tarpeen kokemus. Tämä on mahdollista saavuttaa kuluttamalla toisen toimijan tarjoama palvelu eli *vastaanottamalla palvelu*. Onnistuneen palvelun vastaanottamisen seurauksena agentilla on se mitä se koki olevansa vailla, tai vastaavasti ei ole sitä mistä se halusi päästä eroon. Onnistunut palvelun vastaanottaminen siirtää agentin pois vastaanottamista edeltäneestä tilasta sellaiseen, jossa tarpeen kokemusta ei enää ole.

Olion tilassa ajan suhteen tapahtuvaa muutosta kutsutaan prosessiksi [Sow00]. Palvelut ovat siis prosesseja, joissa joku agentti tekee jotakin jonkun agentin tarpeen tyydyttämiseksi. Niinpä esimerkiksi tietokoneen korjaaminen itsessään ei ole palvelu. Tietokoneen korjaaminen toisen toimijan tarpeen tyydyttämiseksi on palvelu. Mikä tahansa prosessi voi siis olla palvelu, kunhan siihen liittyy tavoitteena toisen toimijan tarpeen tyydyttäminen.

---

<sup>14</sup><http://sty.stakes.fi/FI/luokituskeskus/palveluluokitus/index.htm>

Myös STAKES-tietoyhteiskuntayksikön terveydenhuollon tietojärjestelmien suunnittelua varten laatimat määritelmät käsitteille *palvelu* ja *palvelutapahtuma* vastaavat kokonaisuutena pitkälti edellä esitettyä määritelmää käsitteelle palvelu. Sosiaali- ja terveydenhuollon asiakas- ja potilasasiakirjasanastossa korostetaan palvelua prosessina, johon liittyy tavoitteena tarpeen tyydyttäminen: ”Organisoidun toiminnan tuloksena syntyvä aineeton hyödyke tarpeiden tyydyttämiseksi” [Vie02a]. STAKES-tietoyhteiskuntayksikön sosiaali- ja terveydenhuollon tietojärjestelmien suunnittelua varten käsite palvelutapahtuma on määritelty siten, että siinä tuodaan esille myös se, että palvelu on kognitiivisten agenttien välinen vuorovaikutusprosessi: ”Palveluntuottajan tai palvelunantajan ja asiakkaan välinen yksittäinen vuorovaikutustilanne...” [Vie02b].

## 7.2 STAKES-palveluluokitus ja sen rajoitukset

STAKES-palveluluokituksen tavoitteena on tarjota valtakunnallisesti yhtenäinen sosiaali- ja terveydenhuollon (soster-huollon) toimialan palveluluokitus, joka kattaa julkisen, yksityisen ja kolmannen sektorin tarjoamat palvelut. Tällainen yhtenäinen valtakunnallisesti sovittu palveluluokitus helpottaisi muiden muassa asiakkaan ja ammattilaisen tiedonhakua ja tilastointia [Oja05b].

STAKES-palveluluokitus sisältää noin 550 nimettyä palvelukäsitettä, jotka on järjestetty hierarkiaksi. STAKES-palveluluokituksen hierarkia on oleellisesti kolmitasoinen. Se määrittelee soster-huollon palvelut yleisellä tasolla, jota toimijat voivat tarkentaa tarvitsemillaan käsitteistöillä [Oja05a].

Ylimmällä tasolla palvelut on jaettu noin 50:n käsitteen (*pääryhmän*) alle. Pääryhmiä ovat esimerkiksi suun terveystalvelut, lääkäripalvelut, hoitotyö, kuntoutus ja tutkimukset. Ne kuvaavat ”palvelualaa”. Pääryhmät jakautuvat alakäsitteiksi, jotka jakautuvat edelleen alakäsitteiksi tarkentaen ylemmän tason määritelmää. Hierarkian muodostamisen periaatteena on ollut se, että alakäsitteet perivät kaikki yläkäsitteen piirteet ja lisäävät vähintään yhden lisäpiirteen yläkäsitteensä määritelmään [Oja05a]. Tarkentavana piirteenä voi olla kuvailu palvelun kohteesta, toteuttajasta, toteutustavasta (muodosta), tarpeesta tai sijaintipaikasta.

Esimerkiksi pääryhmä suun terveystalvelut alakäsitteitä ovat hammaslääkäripalvelut, erikoishammaslääkäriin palvelut, erikoishammaslääkäriin päivystys, erikoishammaslääkäriin konsultaatio, suuhygienistin palvelut ja hammashoitajan palvelut. Kaikki edellä luetellut alakäsitteet lisäävät pääryhmän määritelmään kuvailun suun terveystalvelun toteuttajasta: hammaslääkäri, erikoishammaslääkäri, hammashoitaja tai suuhygienisti. Asiantuntijalle ne voivat samalla lisätä pääryhmän määritelmään myös kuvailun esimerkiksi palvelun tarpeesta. Käsitteet erikoishammaslääkäriin päivystys ja erikoishammaslääkäriin konsultaatio lisäävät suun terveystalvelut käsitteen määritelmään kuvauksen palvelun toteuttamistavasta: päivystys tai konsultaatio.

Suun terveystalvelut käsitteen alakäsite erikoishammaslääkäriin päivystys jakautuu edelleen esimerkiksi alakäsitteisiin 'päivystys - lasten hammashoito', 'päivystys - suu- ja leukakirurgia'. Käsite 'päivystys - lasten hammashoito' lisää käsitteen erikoishammaslääkäriin päivystys määritelmään kuvailun palvelun kohteena olevasta ikäryhmästä, kun taas käsite 'päivystys - suu- ja leukakirurgia' lisää käsitteen erikoishammaslääkäriin päivystys määritelmään tarkennuksen palvelun tarpeesta.

Valtakunnallisen palveluluokituksen on oltava kattava ja yleiskäyttöinen. Palveluntarjoajien erilaiset toimintatavat ja luokituksen käyttö esimerkiksi tiedonhaussa ja tilastoinnissa asettavat kukin runsaasti erilaisia vaatimuksia palveluluokitukselle [Oja05b]. Nykyisen toteutuksen ongelmana on riittävän yleiskäyttöisyyden tai joustavuuden aikaansaaminen. STAKES-palveluluokitus tukee esimerkiksi tiedonhakua ja tilastointia ainoastaan siinä eksplisiittisesti luetelluilla käsitteillä. Tiedonhaussa tai tilastoinnissa voi kuitenkin syntyä tarve käsitteelle, jota ei löydy STAKES-palveluluokituksesta, mutta joka on sen kanssa semanttisesti päällekkäinen.

Tilastoinnissa tai tiedonhaussa voi esimerkiksi syntyä tarve etsiä kaikki lapsiin kohdistuvat soster-huollon palvelut. Tällaisia palveluita on luokiteltuna STAKES-palveluluokituksessa useita, esimerkiksi lastenhoito, vastaanotto - lasten hammashoito, erikoislääkäriin vastaanotto - lastenpsykiatria ja leikkaustoiminta - lastenkirurgia. Siinä ei kuitenkaan ole yhtä käsitettä, jonka avulla olisi mahdollista löytää kaikki lapsiin kohdistuvat palvelut. Ne on ihmisvoimin etsittävä STAKES-palveluluokituksesta, tulkitsemalla sen sisältämiä käsitteitä ja määrittelemällä täsmäyvät käsitteet käsitteeseen 'lapsiin kohdistuva' merkitykseltään sisältyviksi. Palvelun kohdentumisen

lisäksi tilastoinnissa ja tiedonhaussa voi syntyä tarve löytää kaikki palvelut sen tekijän, ajan tai paikan mukaan. Mikä tahansa palveluihin liittyvä piirre voi toimia tarvittavana hakukriteerinä.

Edellä kuvatun ongelman taustalla oleva yleisempi ongelma on se, että vaikka eri toimijoiden (esimerkiksi tilastojien) toimijakohtaiset käsitteistöt kuvailisivat monelta osin samoja tai osin päällekkäisiä kategorioita kuin STAKES-palveluluokitus, niin tätä ei pystytä automaattisesti tunnistamaan. Käsitteistöjen semanttiset päällekkäisyydet joudutaan määrittelemään ihmisvoimin. Tämä on työlästä ja virheeltistä. Relevantteja semanttisia yhteyksiä jää helposti määrittelemättä. Vastaavasti muutosten hallinta on työlästä ja virheeltistä ihmisvoimin tehtynä.

### 7.3 Määriteltävät palvelukäsitteet

STAKES-palveluluokituksesta määriteltäviksi käsitteiksi valittiin muutamia semanttisesti toisiinsa läheisesti liittyviä käsitteitä, jotta jäljempänä voitiin tarkastella niiden avulla käsitelmäärittelyiden väliltä automaattisesti löytyviä semanttisia sisällyssuhteita. Lisäksi tarkasteltavaksi pyrittiin valitsemaan ensisijaisesti käsitteitä, joille on esitetty STAKES-palveluluokituksessa luonnollisen kielen määritelmä. Näin pyrittiin toimimaan, jotta määritelmiä ei jouduttaisi keksimään itse. STAKES-palveluluokituksen 550:stä nimetystä palvelukäsitteestä kuitenkin vain noin 30:lle prosentille on esitetty luonnollisen kielen määritelmä. Jos määriteltäväksi valittiin käsite, jolle STAKES-palveluluokituksessa ei ollut luonnollisen kielen määritelmää, niin tämän käsitteen määritelmän pystyi muodostamaan helposti käsitteen nimen perusteella ilman soster-huollon alan asiantuntemusta.

Keskeisimmän määriteltävien käsitteiden ryhmän muodostavat lapsiin kohdistuvat palvelut. Määriteltäviksi valittiin käsitteet Lastenneuvolapalvelu ja Leikkitoiminta, joille STAKES-palveluluokituksessa oli luonnollisen kielen määritelmä ja Leikkaus-toiminta - Lastenkirurgia, joka oli mahdollista määritellä nimen perusteella ilman terveydenalan asiantuntemusta. Lisäksi määriteltiin lapsiin kohdistuvalta vaikuttava palvelu Lasten kotihoidon tuki. Tämän käsitteen avulla havainnollistettiin muiden muassa virheitä, joita NLP-menetelmiin perustuvat haut helposti tekevät.

Toisen semanttisesti läheisen toisiinsa liittyvien käsitteiden joukon muodostavat käsitteet sairaankuljetus (*Sairaankuljetus*) ja ensihoito (*Ensihoito*). Nämä ovat ensimmäiset STAKES-palveluluokituksesta löytyvät käsitteet, joille on esitetty luonnollisen kielen määritelmä.

## 7.4 Palvelukäsitteiden määrittely OWL+SUMO-menetelmällä

Seuraavaksi on tarkoitus tarkastella esimerkein, kuinka toimijat voivat kuvauslogiikka+primitiivikäsitteistö-menetelmällä määrittellä toisistaan riippumattomasti toimialakohtaisia käsitteitä. Esimerkkien käsittemäärittelyt tehdään käyttäen kuvauslogiikkana OWL-kuvauslogiikkaa ja primitiivikäsitteistönä SUMO-yläontologiaa. Niinpä tähän käytettävään menetelmään viitataan nimellä OWL+SUMO. Määrittelyissä käytetään kuvauslogiikan syntaksia ja selkeyden vuoksi olioihin viitataan termeillä eikä webissä käytettävillä yksilöivillä URI-tunnisteilla. Käsitteiden määrittelyt perustetaan STAKES-palveluluokituksessa niille esitettyihin nimiin tai luonnollisen kielen määritelmiin.

Seuraavien esimerkkien tavoitteena on menetelmän esittelyn lisäksi tarjota systemaattisesti esiteltyjä esimerkkimäärittelyjä. Käsittemäärittelyjen yhteydessä tarkastellaan myös lähemmin SUMO-ontologiassa määriteltyjä käsitteitä. Käsittemäärittelyjen jälkeen tarkastellaan ja testataan esimerkkimäärittelyjen avulla, kuinka toimijat voivat löytää OWL+SUMO-menetelmällä määriteltyjen käsitteiden välisiä semanttisia suhteita automaattisesti, olemassa olevilla päättelykoneilla.

SUMO-ontologiassa ei ole käsitettä palvelu. Tämä on selkeä puute, koska palvelun käsitettä on erittäin vaikea määrittellä SUMO-ontologian käsitteiden avulla. Aina-kin jos formalisoitavana kategoriana on ”prosessit, joissa joku agentti tekee jotakin jonkun agentin tarpeen tyydyttämiseksi”. Erityisesti ongelmana on kuvailla käsite ”tarpeen tyydyttäminen”.

Semanttinen yhteentoimivuus on mahdollista, jos kaikki määrittelyt perustuvat primitiivikäsitteistöön. Lisäksi semanttinen yhteentoimivuus toimii sitä paremmin, mitä täsmällisemmin käsitteet pystytään määrittelemään primitiivikäsitteiden avulla. Palvelulle joudutaan kuitenkin tekemään vain *osittainen* (engl. partial) eli ainoas-



taan välttämättömistä ehdoista koostuva määritelmä, koska kategoriaan kuulumisen riittäviä ehtoja palvelulle, *täydellisen* (engl. complete) määritelmän muotoilemiseksi, on erittäin vaikea tai mahdotonta määritellä SUMO-ontologian käsitteiden avulla.

SUMO-ontologiassa on edustettuna käsite ”kaupallinen palvelu” (**CommercialService**)<sup>15</sup>. Se on määritelty kaupallisen toimijan (**CommercialAgent**) suorittamaksi taloudelliseksi transaktioksi (**FinancialTransaction**), jonka tavoitteena on voiton tuottaminen. Kaupallisen palvelun epäsuorana yläkäsitteenä SUMO-ontologiassa on sosiaalinen vuorovaikutus (**SocialInteraction**). Tämä käsite on määritelty intentionaalisen prosessin (**IntentionalProcess**) alakäsitteeksi, jossa kaksi kognitiivista toimijaa (**CognitiveAgent**) ovat keskenään vuorovaikutuksessa.

Jos palvelu määritellään prosessiksi, jossa jokin toimija pyrkii tyydyttämään toisen toimijan tarpeen, niin SUMO-ontologian käsitteiden avulla yleisenkin palvelun käsitteen merkitys voidaan määritellä sisältyväksi ainakin käsitteen **IntentionalProcess** ja edelleen käsitteen **SocialInteraction** merkitykseen:

$$\text{Palvelu} \sqsubseteq \text{SocialInteraction.}$$

Käsitteitä määriteltäessä selkeästi koosteiset käsitteet on hyvä jakaa osiin, jotka nimetään ja määritellään omina kokonaisuuksinaan. Lopuksi näistä nimetyistä kokonaisuuksista koostetaan osituksen kohteena ollut koosteinen käsite. Koosteisten käsitteiden osittaminen nimetyiksi määritelmiksi mahdollistaa määrittelyiden uudelleenkäytön uusia käsitteitä määriteltäessä. Tämä yhdenmukaistaa toimijan tuottamia määritelmiä, mikä parantaa semanttista yhteentoimivuutta.

Esimerkiksi STAKES-Palveluluokituksessa kuvailtu käsite **Ensihoito** on määritelty seuraavasti: ”Alan koulutuksen saaneen ammattihenkilön antama hoito tapahtumapaikalla tai ennen hoitopaikkaan saapumista”. Tämä käsite koostuu selvästi primitiivisemmistä käsitteistä ”alan koulutuksen saaneen ammattihenkilön antama hoito”, tapahtumapaikka ja ”ennen hoitopaikkaan saapumista”. Näitä käsitteitä on helppo

---

<sup>15</sup>Lihavoidulla kirjasintyyllillä esitetyt termit viittaavat SUMO-ontologiassa esitettyihin primitiivikäsitteisiin tai ominaisuuksiin ja teletype-kirjasintyyllillä esitetyt termit viittaavat määriteltyihin käsitteisiin tai ominaisuuksiin. Käsitteiden nimet alkavat isolla alkukirjaimella ja ominaisuuksien nimet alkavat pienellä alkukirjaimella.

jakaa edelleen primitiivisempiin käsitteisiin: hoito, alan koulutuksen saanut ammattihenkilö, hoitopaikka ja niin edelleen. Käsitteiden ositusta jatketaan kunnes päädytään primitiivikäsitteisiin tai niiden avulla suoraviivaisesti määriteltäviin käsitteisiin. Esimerkiksi käsitettä hoito vastaa SUMO-ontologiasta löytyvä primitiivikäsite **TherapeuticProcess**. Tämän avulla voidaan kuvata käsite ”alan koulutuksen saaneen ammattihenkilön antama hoito” (**Ammattihoito**). Ensin määritellään kuitenkin sen sisältämä käsite ”alan koulutuksen saanut ammattihenkilö”.

STAKES-Palveluluokituksessa esitetystä käsitteen **Ensihoito** määritelmästä ei käy eksplisiittisesti ilmi, minkä alan koulutuksen saaneesta ammattihenkilöstä on kyse. Tässä käsite **Ammattihoito** oletetaan nimenomaan hoitoalan koulutuksen saaneen ammattihenkilön (**HoitoalanAmmattihenkilö**) antamaksi hoidoksi. SUMO-ontologiassa käsitteelle prosessi (**Process**) on määritelty ominaisuus taitaja (**hasSkill**)<sup>16</sup>, jonka arvo tulkitaan toimijaksi (**Agent**). Yleisellä eksistentiaalisella kvantifioinnilla tämän ominaisuuden avulla käsitteestä **TherapeuticProcess** voidaan muodostaa käsite, joka tulkitaan hoitamisen taidon omaaviksi olioiksi. Muodostamalla leikkaus tämän ja käsitteen ihminen välillä muodostuu käsite, joka tulkitaan hoitamisen taidon omaaviksi henkilöiksi. Tämä on selkeästi helpoin tapa päästä SUMO-ontologian käsitteiden avulla melko lähelle käsitteen **HoitoalanAmmattihenkilö** merkitystä.

SUMO-ontologiassa on edustettuna käsite ihminen (**Human**), mutta sille ei ole määritelty ominaisuutta **hasSkill** vastaavaa ominaisuutta. Muodostettaessa leikkaus käsitteen **Human** kanssa tarkoituksena on määritellä käsite, joka tulkitaan hoitamisen taidon omaavien ihmisten muodostamaksi joukoksi. Toisin sanoen olion taitamien prosessien sijaan on pystyttävä määrittelemään olioita, joilla on hoitamisen taito. Niinpä käsitteen **Human** kanssa tehtävää leikkausta varten **hasSkill** ominaisuudelle on määriteltävä käänteisominaisuus **hasSkillOf**:

$$\text{hasSkillOf} \equiv (\neg \text{hasSkill})$$

Nyt hoitamisen taidon omaaviksi ihmisiksi tulkittava käsite voidaan määritellä seuraavasti:

---

<sup>16</sup>SUMO-ontologiassa ”taitaja” ominaisuuden nimi on nurinkurisesti ”omaa taidon” (**hasSkill**). Esimerkiksi jos on määritelty **Hoito-omaaTaidon-Henkilö**, niin tälle kolmikolle ei saada selkeää mielekästä tulkintaa tavallisella kolmikoiden lukusuunnalla: subjekti-predikaatti-objekti.

**HoitoalanAmmattihenkilö**  $\equiv$  **Human**  
 $\sqcap (\exists \text{hasSkillOf}.\text{TherapeuticProcess}).$

Käsite **TherapeuticProcess** on SUMO-ontologiassa epäsuora alakäsite käsitteelle **Process**, jolta se perii muiden muassa ominaisuudet tekijä (**agent**) ja kohde (**patient**). Ominaisuuden **agent** arvo ilmentää prosessin aktiivista määräävää tekijää ja ominaisuuden **patient** arvo ilmentää prosessin kohdetta. Yleisellä eksistentiaalisella kvantifioinnilla **agent** ominaisuuden avulla käsitteestä **HoitoalanAmmattihenkilö** voidaan muodostaa käsite, joka tulkitaan hoitoalan ammattihenkilön suorittamien prosessien joukoksi:

$\exists \text{agent.HoitoalanAmmattihenkilö}.$

Muodostamalla leikkaus hoidon ja hoitoalan ammattihenkilön suorittamien entiteettien<sup>17</sup> välillä muodostuu käsite ammattihoito, joka tulkitaan hoitoalan ammattihenkilön suorittamien hoitojen joukoksi:

**Ammattihoito**  $\equiv$  **TherapeuticProcess**  
 $\sqcap (\exists \text{agent.HoitoalanAmmattihenkilö})$

Käsitteen **Ensihoito** määrittämiseksi on vielä määriteltävä sen sisältämät primitiivisemmät käsitteet ”tapahtumapaikka” ja ”ennen hoitopaikkaan saapumista”. Formalisoitaessa alkuperäisiä STAKES-luokituskeskuksen kielellisin ilmauksin kuvailtuja määritelmiä voidaan selvästi havaita, että ne jättävät paljon tulkinnan varaa. Tapahtumapaikka ja ennen hoitopaikkaan saapumista ovat käsitteinä hyvin laiveita määrittelemään paikkaa ja aikaa, jolloin ensihoitoa tyypillisesti annetaan. Käsite tapahtumapaikka ei oikeastaan rajaa hoidon paikkaa mitenkään. Käsite ennen hoitopaikkaan saapumista ei taas kiinnitä hoidon alkamiskohtaa millään tavalla. Ensihoidon paikkaa kuvaavampi ja täsmällisempi määritelmä voisi olla esimerkiksi fysiologisen vahingon tapahtumapaikka, tapaturmapaikka (**Tapaturmapaikka**). Vastaavasti

---

<sup>17</sup>SUMO-ontologiassa kaikkien mahdollisten olevaisten eli olioiden joukkoon viittaava käsite on nimetty entiteetiksi (engl. entity). Termit olio ja entiteetti ovat siis synonyymeja.

ensihoidon ajankohtaa kuvaavampi ja täsmällisempi määritelmä voisi olla esimerkiksi hoitopaikkaan siirtymisen aikainen ajanjakso (**HoitopaikkaanSiirtymisenAika**). SUMO-ontologiassa kaikille fysikaalisille (**Physical**) on määritelty ominaisuus sijaitsee (**located**), jonka arvo tulkitaan aineelliseksi kohteeksi (**Object**). Käsite **Physical** tulkitaan entiteetiksi (**Entity**), jolla on sijainti ajassa ja avaruudessa. Käsitteen **Entity** ekstensioon kuuluvat kaikki mahdolliset yksilöt. Käsitteen **Object** ekstensioon kuuluvat kaikkien aineellisten kohteet, alueet mukaan luettuna. Ominaisuudella **located** voidaan siis määritellä fysikaalisten olioiden sijainteja aineellisissa kohteissa tai alueilla. Käsitteen tapaturmapaikka määrittely vaatii ominaisuuden, jolla on mahdollista määritellä aineellisia kohteita fysikaalisten olioiden sijaintipaikkoina. Käsitteen tapaturmapaikka määrittely vaatii siis ominaisuuden **located** käänteisominaisuutta **locationOf**:

$$\text{locationOf} \equiv (\neg\text{located}).$$

SUMO-ontologiasta löytyy käsitteet vahingoittuminen (**Injuring**) ja paikka tai alue (**Region**). Käsite (**Injuring**) määritellään organismin häiriintyneeksi prosessiksi tai tilaksi. Yleisellä eksistentiaalisella kvantifioinnilla **locationOf** ominaisuuden avulla käsitteestä **Injuring** voidaan muodostaa käsite, joka tulkitaan vahingoittumisen tapahtumapaikkana toimivien fysikaalisten olioiden joukoksi. Muodostamalla leikkaus tämän ja käsitteen **Region** välille muodostuu käsite, joka tulkitaan vahingon tapahtumapaikkana toimivien alueiden joukoksi eli tapaturmapaikaksi (**Tapaturmapaikka**):

$$\text{Tapaturmapaikka} \equiv \text{Region} \sqcap (\exists\text{locationOf.Injuring})$$

Joissakin tapauksissa voi olla selkeämpää edetä määrittelyssä koosteisemmista käsitteistä primitiivisempiin. Edellä käsitteen **Ensihoito** määrittelemisen aloitettiin primitiivisemmistä käsitteistä kohti koosteisempia käsitteitä. **Ensihoito**-käsitteen sisältämän käsitteen **HoitopaikkaanSiirtymisenAika** määrittelyssä edetään toisinpäin.

SUMO-ontologiassa fysikaalisille on määritelty ominaisuus aika (**time**), jolla voidaan määritellä jonkin ajanjakson aikana esiintyviä fysikaalisia. Käsitteen **HoitopaikkaanSiirtymisenAika** määrittely vaatii kuitenkin ominaisuuden, jolla on mah-

dollista määritellä ajanjaksoja, joiden aikana tapahtuu tai esiintyy tiettyjä fysikaalisia. Käsitteen **HoitopaikkaanSiirtymisenAika** määrittely vaatii siis ominaisuuden **time** käänteisominaisuutta **timeOf**:

$$\text{timeOf} \equiv (\neg \text{time}).$$

Yleisellä eksistentiaalisella kvantifioinnilla ominaisuuden **timeOf** avulla käsitteestä **HoitopaikkaanSiirtyminen** voidaan muodostaa käsite, joka tulkitaan hoitopaikkaan siirtymisen kanssa samanaikaisten ajanjaksojen tai ajanhetkien joukoksi. Muodostamalla leikkaus tämän ja käsitteen **TimeInterval** välille muodostuu käsite, joka tulkitaan hoitopaikkaan siirtymisen kanssa samanaikaisten ajanjaksojen joukoksi:

$$\text{HoitopaikkaanSiirtymisenAika} \equiv \\ \text{TimeInterval} \sqcap (\exists \text{timeOf.HoitopaikkaanSiirtyminen}).$$

SUMO-ontologiassa on edustettuna käsite siirtyminen paikasta toiseen (**Translocation**, jolle on määritelty ominaisuudet lähtökohta (tai alkuperä) (**origin**) ja määränpää (**destination**). Ominaisuudella **origin** on mahdollista määritellä jonkin aineellisen kohteen lähtökohtanaan omaavia prosesseja ja ominaisuudella **destination** on vastaavasti mahdollista määritellä jonkin aineellisen kohteen tai alueen määränpäänään omaavia prosesseja. Yleisellä eksistentiaalisella kvantifioinnilla **destination** ominaisuuden avulla käsitteestä **Hoitopaikka** voidaan muodostaa käsite, joka tulkitaan hoitopaikan määränpäänään omaavien prosessien joukoksi. Muodostamalla leikkaus tämän ja käsitteen **Translocation** välille muodostuu käsite, joka tulkitaan hoitopaikan määränpäänään omaavien siirtymisien joukoksi:

$$\text{HoitopaikkaanSiirtyminen} \equiv \text{Translocation} \\ \sqcap (\exists \text{destination.Hoitopaikka})$$

Vielä on määrittelemättä käsite **Hoitopaikka**. Sen määrittely onnistuu **located** ominaisuuden käänteisominaisuudella **locationOf**, jolla on siis mahdollista määritellä aineellisia kohteita tai alueita fysikaalisten olioiden sijaintipaikkoina. Yleisellä eksistentiaalisella kvantifioinnilla **locationOf** ominaisuuden avulla käsitteestä **TherapeuticProcess** voidaan muodostaa käsite, joka tulkitaan hoidon tapahtumapaikkana toimivien aineellisten kohteiden tai alueiden joukoksi:

$\text{Hoitopaikka} \equiv \exists \text{locationOf.TherapeuticProcess.}$

Nyt kaikki käsitteen **Ensihoito** määrittelemistä varten tarvittavat käsitteet on määriteltä SUMO-ontologian primitiivikäsitteiden avulla. STAKES-Palveluluokituksessa käsite **Ensihoito** määritellään ”alan koulutuksen saaneen ammattihenkilön antamaksi hoidoksi tapahtumapaikalla tai ennen hoitopaikkaan saapumista”. Edellä määrittelemiemme käsitteiden avulla on mahdollista muodostaa käsite, joka tulkitaan ”hoitamisen taidon omaavien henkilöiden antamaksi hoidoksi tapaturmapaikalla tai hoitoon siirtymisen aikana”. Yleisellä eksistentiaalisella kvantifioinnilla ominaisuuden **time** avulla käsitteestä **HoitopaikkaanSiirtymisenAika** voidaan muodostaa käsite (**HoitopaikkaanSiirtymisenAikainen**), joka tulkitaan hoitopaikkaan siirtymisen aikana esiintyvien fysikaalisten olioiden joukoksi:

$\text{HoitopaikkaanSiirtymisenAikainen} \equiv \exists \text{time.HoitopaikkaanSiirtymisenAika.}$

Yleisellä eksistentiaalisella kvantifioinnilla **located** ominaisuuden avulla käsitteestä **Tapaturmapaikka** voidaan muodostaa käsite (**TapaturmapaikallaOleva**), joka tulkitaan tapaturmapaikalla sijaitsevien tai tapahtuvien fysikaalisten olioiden joukoksi:

$\text{TapaturmapaikallaOleva} \equiv \exists \text{located.Tapaturmapaikka.}$

Muodostamalla yhdiste näiden käsitteiden välille muodostuu käsite (**HoitoonSiirtymisenAikainenTaiTapaturmapaikallaOleva**), joka tulkitaan tapaturmapaikalla sijaitsevien tai tapahtuvien fysikaalisten olioiden tai hoitopaikkaan siirtymisen aikana esiintyvien fysikaalisten olioiden joukoksi:

$\text{HoitoonSiirtymisenAikainenTaiTapaturmapaikallaOleva} \equiv$

$\text{HoitopaikkaanSiirtymisenAikainen} \sqcup \text{TapaturmapaikallaOleva.}$

Muodostamalla leikkaus tämän yhdisteen ja käsitteen **Ammattihoito** välille muodostuu käsite, joka tulkitaan hoitamisen taidon omaavien henkilöiden suorittamien tapaturmapaikalla sijaitsevien tai tapahtuvien hoitojen tai hoitopaikkaan siirtymisen aikana esiintyvien hoitojen joukoksi:

$\text{Ensihoito} \equiv \text{Ammattihoito}$

$\sqcap \text{HoitoonSiirtymisenAikainenTaiTapaTurmapaikallaOleva}$ .

Käsitteen **Ensihoito** määritelmästä saa eksplisiittisesti määriteltyä **Ensihoito-palvelua** (**Ensihoitopalvelu**) kuvailevan käsitteen, jos muodostetaan leikkaus käsitteen **Ensihoito** ja käsitteen **Palvelu** välille:

$\text{Ensihoitopalvelu} \equiv \text{Ensihoito} \sqcap \text{Palvelu}$ .

Näin määritelty käsite tulkitaan hoitamisen taidon omaavien henkilöiden suorittamien tapaturmapaikalla sijaitsevien tai tapahtuvien hoitojen tai hoitopaikkaan siirtymisen aikana esiintyvien hoitopalveluiden joukoksi.

Vastaavalla tavalla kaikista prosesseista saa OWL+SUMO-menetelmällä määriteltyä palvelun muodostamalla leikkaus määrittelyn prosessikäsitteen ja käsitteen palvelu välille. Koska prosessin määrittely palveluprosessiksi on formalisointina edellistä määrittelyä vastaava riippumatta prosessista, jonka kanssa leikkaus muodostetaan, niin jatkossa määriteltävien prosessien muuntaminen palveluiksi jätetään esittämättä.

Seuraavaksi määritellään käsite sairaankuljetus (**Sairankuljetus**). STAKES-Palveluluokituksessa se kuvaillaan seuraavasti: ”terveydenhuollon tai ensihoitoalan koulutuksen saaneen ammattihenkilön tai ammattihenkilöstön suorittama potilaan siirtäminen mm. ambulanssilla tai helikopterilla hoitopaikkaan”. Kuljettamista eli siirtämistä, jonka tekijä ja kohde ovat eri entiteetit, edustaa SUMO-ontologiassa käsite **Transfer**. Käsitteen **Process** epäsuorana alakäsitteenä käsitteelle **Transfer** on määritelty ominaisuudet **agent** ja **patient**, joilla on mahdollista määritellä kuljettamisen tekijä ja kohde. Kuljetuksen määränpää määritellään ominaisuudella **destination**, jonka käsite **Transfer** perii suoralta ylikäsitteeltään **Translocation**.

Yleisellä eksistentiaalisella kvantifioinnilla **agent** ominaisuuden avulla käsitteestä **HoitoalanAmmattihenkilö** voidaan muodostaa käsite, joka tulkitaan hoitoalan ammattihenkilön suorittamien prosessien joukoksi ja **patient** ominaisuuden avulla käsitteestä **Sairas** voidaan muodostaa käsite, joka tulkitaan sairaat kohteenaan omaavien prosessien joukoksi. Yleisellä eksistentiaalisella kvantifioinnilla **destination**

ominaisuuden avulla käsitteestä **Hoitopaikka** voidaan muodostaa käsite, joka tulkitaan hoitopaikan määränpäänään omaavien prosessien joukoksi. Muodostamalla leikkaus näiden ja käsitteen **Transfer** kesken muodostuu käsite, joka tulkitaan hoitamisen taidon omaavien henkilöiden suorittamien hoitopaikan määränpäänään omaavien sairaidenkuljetusten joukoksi:

$$\begin{aligned} \text{Sairaankuljetus} &\equiv \text{Transfer} \\ &\sqcap (\exists \text{agent.HoitoalanAmmattihenkilö}) \\ &\sqcap (\exists \text{patient.Sairas}) \\ &\sqcap (\exists \text{destination.Hoitopaikka}). \end{aligned}$$

Käsitteen **Sairaankuljetus** määrittelyssä esiintyvä käsite **Sairas** on vielä määriteltävä. Käsitteen **Sairaankuljetus** luonnollisen kielen kuvailun perusteella siinä esiintyvälle käsite **Sairas** tulkitaan joiltain osin epänormaalisti toimivaksi tai epänormaalissa tilassa olevaksi ihmiseksi (**Human**). SUMO-ontologiassa tällaista organismin (**Organism**) epänormaalia toimintaa tai tilaa edustaa käsite (**DiseaseOrSyndrome**). SUMO-ontologiassa kaikille entiteeteille on määritelty ominaisuus **property**, jolla voidaan määrittellä jonkin ominaisuuden (**Attribute**) omaavia entiteettejä. Yleisellä eksistentiaalisella kvantifioinnilla tämän ominaisuuden avulla käsitteestä **DiseaseOrSyndrome** voidaan muodostaa käsite, joka tulkitaan sairaiksi entiteeteiksi. Muodostamalla leikkaus tämän ja käsitteen **Human** välillä muodostuu käsite, joka tulkitaan sairaiden ihmisten muodostamaksi joukoksi:

$$\begin{aligned} \text{Sairas} &\equiv \text{Human} \\ &\sqcap (\exists \text{property.DiseaseOrSyndrome}). \end{aligned}$$

Lopuksi määritellään vielä käsitteet: Leikkaustoiminta-Lastenkirurgia (**Leikkaustoiminta-Lastenkirurgia**), Lastenneuvolapalvelu (**Lastenneuvolapalvelu**), Leikkitoiminta (**Leikkitoiminta**) ja Lasten kotihoidon tuki (**LastenKotihoidonTuki**). Nämä käsitteet määritellään edellä esitettyssä järjestyksessä. STAKES-Palveluluokituksessa käsitettä **Leikkaustoiminta-Lastenkirurgia** ei ole kuvailtu tarkemmin. Ilmeisesti sen merkityksen voidaan olettaa olevan sen nimen perusteella itsestään selvä. Muille edellä mainituille käsitteille luokituksesta löytyy täsmällisempi kuvailu.



SUMO-ontologiassa ei ole määritelty käsitettä lapsi (**Lapsi**). STAKES-Palveluluokituksessakaan tälle käsitteelle ei ole kuvailua. Perustamme määrittelymme Medical Subject Headings<sup>18</sup> (MeSH) -käsitteistössä käsitteelle **Lapsi** (MeSH Heading: Child) esitettyyn määritelmään. Siellä lapsi on määritelty iältään kuuden ja 12:n vuoden suljetulle välille kuuluvien ihmisten muodostamaksi joukoksi. Määrittelyn yhteydessä huomautetaan, että iältään kahden ja viiden vuoden suljetulle välille kuuluvat ihmiset muodostavat käsitteen ”alle kouluikäinen lapsi” (**AlleKouluikäinenLapsi**) (MeSH Heading: Child, Preschool) ekstension. Oletettavasti myös alle kaksi vuoti-aat ihmiset ovat lapsia. Ainakin tämä vastaisi arkijärjen tulkintaa käsitteelle **Lapsi**. Niinpä käsite **Lapsi** määritellään tässä, vuoden tarkkuudella ilmaistuna, iältään nollan ja 12:n vuoden suljetulle välille kuuluvien ihmisten muodostamaksi joukoksi.

Käsite **Human** on epäsuora alakäsite käsitteelle **Object**, jolta se perii ominaisuuden ikä (**age**). Ominaisuuden **age** avulla on mahdollista määritellä käsitteen **Object** ekstensioon kuuluvien yksilöiden ikää. Käsitteen **Lapsi** määritelmässä ikä ilmoitetaan vuosina mitattuina ajanpituuksina: yksi vuotta, kaksi vuotta, kolme vuotta ja niin edelleen. SUMO-ontologian näkökulmasta nämä voidaan nähdä vuosina mitattujen ajanpituuksien muodostamaan joukkoon (**YearDuration**) kuuluvina yksilöitä. Käsite **YearDuration** on epäsuora alakäsite käsitteelle **ConstantQuantity**, jolta se perii ominaisuuden **MagnitudeFn**. Tämän ominaisuuden avulla on mahdollista määritellä pituuksien ja muiden määrien suuruus lukuarvona. Käsitteen **Lapsi** määrittelyä MeSH-käsitteistössä esitetyllä tavalla, on pystyttävä määrittelemään käsite (tai tietotyyppi), joka tulkitaan ”välille nolasta kahteentoista kuuluvaksi lukuarvoksi”.

Tietotyyppien tulkinta toteutetaan tavallisesti muusta päättelystä erillisenä palveluna. Osana tätä palvelua tarjotaan myös tavallisesti kuvauslogiikasta erillinen nimenomaan tietotyyppien määrittelyyn tarkoitettu formalismi [W3Cf]. Yksi käytetyimmistä tällaisista formalismeista on määritelty XML Schema Part 2 (XML-SCHEMA2) -suosituksessa [W3Cg], jonka avulla käyttäjän on mahdollista määritellä omia tietotyyppejä. Sen avulla on esimerkiksi helppoa määritellä tietotyyppi, joka tulkitaan nolasta kahteentoista kuuluvien lukuarvojen joukoksi.

---

<sup>18</sup><http://www.nlm.nih.gov/mesh/>

XML-SCHEMA2-suosituksessa on määriteltynä erilaisia *rajoitteita* ja *suhteita* (engl. Constraining facets ja Fundamental facets), joilla voidaan koosteisesti määritellä perustyyppejä (engl. simple type) (tietotyyppejä) tietoarvoista ja *sisäänrakennetuista* (engl. built-in) tietotyypeistä. Esimerkiksi **equal** (vastaavuus), **cardinality** (kardinaliteetti) ja **minExclusive** ovat XML-SCHEMA2-suosituksessa määriteltäviä suhteita ja rajoitteita. Sisäänrakennettuja tietotyyppejä ovat esimerkiksi **string**, **boolean**, **int** ja **positiveInteger** [W3Cg].

Rajoitteella **maxInclusive**, lukuarvolla 12 ja sisäänrakennetulla tietotyypillä **positiveInteger** voidaan määritellä tietotyyppi **alle13**, joka tulkitaan välille nolasta kahteentoista kuuluvien positiivisten kokonaislukujen joukoksi:

```
<simpleType name='''alle13''>
  <restriction base='''positiveInteger''>
    <maxInclusive value='''12''/>
  </restriction>
</simpleType>.
```

Yleisellä eksistentiaalisella kvantifioinnilla ominaisuuden **MagnitudeFn** avulla tietotyyppistä **alle13** voidaan muodostaa käsite, joka tulkitaan suuruudeltaan lukuarvojen kuuden ja 12:n välille kuuluviksi joukoiksi. Määrittelemällä leikkaus tämän ja käsitteen käsitteen **YearDuration** välille saadaan muodostettua käsite **Alle13Vuotta**, joka tulkitaan nollassa ja 12:n vuoden suljetulle välille kuuluvien vuosina mitattavien ajanpituuksien muodostamaksi joukoksi:

$$\text{Alle13Vuotta} \equiv \text{YearDuration} \sqcap (\exists \text{MagnitudeFn.alle13})$$

Yleisellä eksistentiaalisella kvantifioinnilla ominaisuuden **age** avulla käsitteestä **Alle13Vuotta** voidaan muodostaa käsite, joka tulkitaan iältään nollassa ja 12:n vuoden suljetulle välille kuuluvien käsitteen **Object** ekstensioon kuuluvien yksilöiden muodostamaksi joukoksi. Muodostamalla leikkaus tämän ja käsitteen **Human** välillä muodostuu käsite, joka tulkitaan iältään nollassa ja 12:n vuoden suljetulle välille kuuluvien ihmisten muodostamaksi joukoksi:

$$\begin{aligned} \text{Lapsi} &\equiv \text{Human} \\ &\sqcap \leq \text{1age} \\ &\sqcap (\exists \text{age. Alle13Vuotta}) \end{aligned}$$

Ihmisellä ei tietenkään voi olla kuin yksi ikä, joten lapsen iälle on asetettu maximikardinaliteetiksi yksi ( $\leq \text{1age}$ ).

Kirurgista prosessia SUMO-ontologiassa edustaa käsite **Surgery**. Yleisellä eksistentiaalisella kvantifoinnilla **patient** ominaisuuden avulla käsitteestä **Lapsi** voidaan muodostaa käsite, joka tulkitaan lapset kohteenaan omaavien prosessien muodostamaksi joukoksi. Muodostamalla leikkaus tämän ja käsitteen **Surgery** välille muodostuu käsite, joka tulkitaan lapset kohteenaan omaavien leikkausprosessien muodostamaksi joukoksi:

$$\begin{aligned} \text{Leikkaustoiminta-Lastenkirurgia} &\equiv \text{Surgery} \\ &\sqcap (\exists \text{patient.Lapsi}). \end{aligned}$$

STAKES-Palveluluokituksessa **Lastenneuvolapalvelu** kuvaillaan ”alle kouluikäisille lapsille ja vanhemmille annettavaksi terveysneuvonnaksi, joka sisältää myös lapsen tutkimus- ja rokotuspalveluja”. Myös ”alle kouluikäiset lapset” (**AlleKouluikäinenLapsi**) käsitteen määrittely perustetaan MeSH-käsitteistössä esitettyyn määrittelmään (iältää kahden ja viiden vuoden suljetulle välille kuuluvat ihmiset), joka laajennetaan kattamaan myös alle kaksi vuotiaat ihmiset. Formalisointina käsitteen **AlleKouluikäinenLapsi** määrittely vastaa käsitteen **Lapsi** määrittelyä.

Aluksi määritellään XML-SCHEMA2-suosituksella tietotyyppi **alle6**:

```
<simpleType name='''alle6''>
  <restriction base='''positiveInteger''>
    <maxInclusive value='''5''/>
  </restriction>
</simpleType>.
```

eli rajoitteella **maxInclusive**, lukuarvolla 5 ja sisäänrakennetulla tietotyypillä **positiveInteger** määritellä tietotyyppi, joka tulkitaan välille nolasta viiteen kuuluvien positiivisten kokonaislukujen joukoksi. Tämän tietotyypin avulla määritellään käsite alle kuusi vuotta (**Alle6Vuotta**):

$$\begin{aligned} \text{Alle6Vuotta} &\equiv \text{YearDuration} \\ &\sqcap (\exists \text{MagnitudeFn.alle6}). \end{aligned}$$

Lopuksi ominaisuuden **age**, edellä määritellyn käsitteen **Alle6Vuotta** ja käsitteen **Human** avulla määritellään käsite **AlleKouluikäinenLapsi**:

$$\begin{aligned} \text{AlleKouluikäinenLapsi} &\equiv \text{Human} \\ &\sqcap \leq \text{lage} \\ &\sqcap (\exists \text{age. Alle6Vuotta}). \end{aligned}$$

Käsitteelle **Organism** on määritelty ominaisuus **parent**. Sen avulla voidaan määritellä organismeja, joilla on tietynlainen organismi vanhempana. Käsitteen **AlleKouluikäisenVanhempi** määrittelyä varten tarvitaan kuitenkin käsite, jolla voidaan määritellä tietynlaisen organismin lapsenaan omaavia organismeja. Tämän määrittelyä varten käsitteelle **parent** on määriteltävä käänteisominaisuus:

$$\text{parentOf} \equiv (\neg \text{parent}).$$

Yleisellä eksistentiaalisella kvantifiointilla **parentOf** ominaisuuden avulla käsitteestä **AlleKouluikäinenLapsi** voidaan muodostaa käsite, jonka käsitteen **Human** kanssa muodostama leikkaus tulkitaan alle kouluikäisten lasten omaavien ihmisten muodostamaksi joukoksi:

$$\begin{aligned} \text{AlleKouluikäisenVanhempi} &\equiv \text{Human} \\ &\sqcap (\exists \text{parentOf. AlleKouluikäinenLapsi}). \end{aligned}$$

SUMO-ontologiassa on määriteltynä prosessia opettaminen edustava käsite (**EducationalProcess**). Tämän avulla on mahdollista määritellä käsitteen **Lastenneuvola** sisältämä primitiivisempi käsite **terveysneuvonta** (**Terveysneuvonta**).

Käsitteelle **EducationalProcess** on määritelty ominaisuus **represents**, jolla voidaan määritellä jotakin entiteettiä edustavia, esitteleviä tai kuvailevia entiteettejä. SUMO-ontologiassa ei ole edustettuna käsitettä **terveys**, eikä STAKES-Palveluluokituksessa ole määritelty täsmällisemmin sitä, mitä käsite **terveys** tarkoittaa terveysneuvonnan yhteydessä. Käsite **Terveysneuvonta** määritellään tässä hieman laveasti organismin biologisia ominaisuuksia (**BiologicalAttribute**) esitteleväksi opetuksiksi.

Yleisellä eksistentiaalisella kvantifoinnilla **represents** ominaisuuden avulla käsitteestä **BiologicalAttribute** voidaan muodostaa käsite, jonka käsitteen **EducationalProcess** kanssa muodostama leikkaus tulkitaan organismin biologisia ominaisuuksia esittelevien opetusprosessien joukoksi:

$$\begin{aligned} \text{Terveysneuvonta} &\equiv \text{EducationalProcess} \\ &\sqcap (\exists \text{represents.BiologicalAttribute}). \end{aligned}$$

Käsite ”alle kouluikäiselle lapselle tai alle kouluikäisen lapsen vanhemmalle kohdistuva terveysneuvonta” **AlleKouluikäisenTaiSellaisenVanhemmanTerveysneuvonta** määritellään omaksi nimetyksi kokonaisuudeksi. Tämän määrittelemiseksi muodostetaan ensin sen sisältämät primitiivisemmät käsitteet: alle kouluikäisen terveysneuvonta **AlleKouluikäisenTerveysneuvonta** ja alle kouluikäisen lapsen vanhemman terveysneuvonta **AlleKouluikäisenVanhemmanTerveysneuvonta**.

Yleisellä eksistentiaalisella kvantifoinnilla **patient** ominaisuuden avulla käsitteestä **AlleKouluikäinenLapsi** voidaan muodostaa käsite, joka tulkitaan alle kouluikäiseen lapseen kohdistuvien prosessien joukoksi. Muodostamalla leikkaus tämän ja käsitteen **Terveysneuvonta** välille muodostuu käsite, joka tulkitaan alle kouluikäiset lapset kohteenaan omaavien terveysneuvonta prosessien muodostamaksi joukoksi:

$$\begin{aligned} \text{AlleKouluikäisenTerveysneuvonta} &\equiv \text{Terveysneuvonta} \\ &\sqcap (\exists \text{patient.AlleKouluikäinenLapsi}). \end{aligned}$$

Vastaavalla tavalla **patient** ominaisuuden avulla muodostetaan käsite, joka tulkitaan alle kouluikäisten lasten vanhemmat kohteenaan omaavien terveysneuvonta prosessien muodostamaksi joukoksi:

$$\text{AlleKouluikäisenVanhemmanTerveysneuvonta} \equiv \text{Terveysneuvonta} \\ \sqcap (\exists \text{patient. AlleKouluikäisenVanhempi}).$$

Näiden käsitteiden muodostama yhdiste tulkitaan alle kouluikäiset lapset tai alle kouluikäisten lasten vanhemmat kohteenaan omaavien terveysneuvonta prosessien muodostamaksi joukoksi:

$$\text{AlleKouluikäisenTaiSellaisenVanhemmanTerveysneuvonta} \equiv \\ \text{AlleKouluikäisenTerveysneuvonta} \\ \sqcup \text{AlleKouluikäisenVanhemmanTerveysneuvonta}.$$

Käsitteet ”biologisesti aktiivisen aineen ruiskuttaminen jonkin ihmisen tai eläimen sisään” (**Injecting**) ja tutkiminen (**Investigating**) ovat edustettuina SUMO-ontologiassa. Yleisellä eksistentiaalisella kvantifioinnilla **patient** ominaisuuden avulla näistä primitiivikäsitteistä voidaan muodostaa käsitteet, jotka tulkitaan vastaaviksi alle kouluikäisiin lapsiin kohdistuviksi prosesseiksi:

$$\text{AlleKouluikäisenRokotuspalvelu} \equiv \text{Injecting} \\ \sqcap (\exists \text{patient. AlleKouluikäinenLapsi})$$

ja

$$\text{AlleKouluikäisenTutkimuspalvelu} \equiv \text{Investigating} \\ \sqcap (\exists \text{patient. AlleKouluikäinenLapsi}).$$

Muodostamalla yhdiste käsitteiden **AlleKouluikäisenTaiSellaisenVanhemmanTerveysneuvonta**, **AlleKouluikäisenRokotuspalvelu** ja **AlleKouluikäisenTutkimuspalvelu** välille muodostuu käsite, joka tulkitaan alle kouluikäisille tai sellaisen vanhemmille suunnattujen terveysneuvontojen joukoksi tai lasten rokotus- tai tutkimuspalveluksi:

$$\text{Lastenneuvolapalvelu} \equiv \\ \text{AlleKouluikäisenTaiSellaisenVanhemmanTerveysneuvonta} \\ \sqcup \text{AlleKouluikäisenRokotuspalvelu} \\ \sqcup \text{AlleKouluikäisenTutkimuspalvelu}.$$

STAKES-Palveluluokituksessa **Leikkitoiminta** kuvaillaan seuraavasti: ”Lapsille suunnattu leikin ja toiminnan ohjausta ja valvontaa sisältävä lasten päivähoito”. Määritelmästä ei suoraan ilmene mitä ”päivähoidolla” täsmällisesti ottaen tarkoitetaan. Jotta tässä olisi mahdollista esitellä OWL+SUMO-menetelmän soveltamista mahdollisimman monipuolisesti, niin tulkitsemme käsitteen ”päivähoito” arkipäivisin tapahtuvaksi lastenhoidoksi. Leikkitoiminta on siis lasten huolehtimista ja heidän leikkien ohjaamista ja valvontaa arkipäivisin.

Leikki (**Leikki**) määritellään SUMO-ontologian avulla lapsen ajanvietteeksi. SUMO-ontologiassa on käsite **RecreationOrExercise**, joka tulkitaan ajanvietteiden joukoksi. Yleisellä eksistentiaalisella kvantifioinnilla **agent** ominaisuuden avulla käsitteestä **Lapsi** voidaan muodostaa käsite, jonka käsitteen **RecreationOrExercise** kanssa muodostama leikkaus tulkitaan lasten suorittamien ajanvietteiden joukoksi:

$$\text{Leikki} \equiv \text{RecreationOrExercise} \\ \sqcap (\exists \text{agent.Lapsi}).$$

SUMO-ontologiassa on edustettuna käsite ohjaus (**Guiding**), joka tulkitaan minä tahansa käsitteen **Object** ekstensioon kuuluvan yksilön käyttäytymisen ohjailujen muodostamaksi joukoksi. Yleisellä eksistentiaalisella kvantifioinnilla **patient** ominaisuuden avulla käsitteestä **Leikki** voidaan muodostaa käsite, jonka käsitteen **Guiding** kanssa muodostama leikkaus tulkitaan lasten suorittamien ajanvietteiden ohjailujen muodostamaksi joukoksi:

$$\text{LeikinOhjaus} \equiv \text{Guiding} \\ \sqcap (\exists \text{patient.Leikki}).$$

SUMO-ontologian käsite **Maintaining** tulkitaan huolehtimiseksi, valvomiseksi ja ylläpitämiseksi. Yleisellä eksistentiaalisella kvantifioinnilla **patient** ominaisuuden avulla käsitteestä **Lapsi** voidaan muodostaa käsite, jonka käsitteen **Maintaining** kanssa muodostama leikkaus tulkitaan lastenhoitojen muodostamaksi joukoksi:

$$\text{Lastenhoito} \equiv \text{Maintaining} \\ \sqcap (\exists \text{patient.Lapsi}).$$

Leikkitoiminnan ajankohdan määrittelemiseksi on määriteltävä käsite arkipäivisin eli maanantaista perjantaihin. Viikonpäiviä edustavat käsitteet (**Monday**, **Tues-**

**day** ja niin edelleen) on SUMO-ontologiassa määritelty käsitteen päivä **Day** alakäsitteiksi. Käsite **Day** on määritelty alakäsitteeksi käsitteelle **TimeInterval**, jolta myös viikonpäiviä edustavat käsitteet perivät ominaisuudet aloittaa **starts** ja lopettaa **finishes**. Ominaisuudella **starts** voidaan määrittellä jonkin ajanjakson aloittavia ajanjaksoja ja ominaisuudella **finishes** voidaan määrittellä jonkin ajanjakson lopettavia ajanjaksoja. Jos ajanjakso  $t_1$  aloittaa ajanjakson  $t$ , niin ajanjakso  $t$  tulkitaan alkavaksi ajanjakson  $t_1$  alkuhetkellä. Jos ajanjakso  $t_2$  lopettaa ajanjakson  $t$ , niin ajanjakso  $t$  tulkitaan loppuvaksi ajanjakson  $t_2$  loppuhetkellä. Käsitteen maanantaista perjantaihin **MaanantaistaPerjantaihin** määrittelemiseksi tarvitaan ominaisuudet, joilla voidaan määrittellä tietyn ajanjakson aloittama ja tietyn ajanjakson lopettama ajanjakso. Niinpä määrittelemme ominaisuuksille **starts** ja **finishes** käänteisominaisuudet aloittava **startsBy** ja lopettava **finishesBy**:

$$\text{startsBy} \equiv (\neg \text{starts})$$

ja

$$\text{finishesBy} \equiv (\neg \text{finishes}).$$

Viikonpäiviä edustavat käsitteet perivät käsitteeltä **TimeInterval** myös käsitteen aikana (**during**), jolla voidaan määrittellä jonkin ajanjakson aikana esiintyvien ajanjaksojen joukkoja. Tämän ominaisuuden avulla voidaan siis määrittellä, että kyse on nimenomaan viikon eikä kuukauden aikana esiintyvien maanantaista perjantaihin kestävien ajanjaksojen joukoista.

Yleisellä eksistentiaalisella kvantifoinnilla **startsBy** ominaisuuden avulla käsitteestä **Monday** voidaan muodostaa käsite, joka tulkitaan ajanjakson maanantai aloittamien ajanjaksojen joukoksi ja vastaavasti **finishesBy** ominaisuuden avulla käsitteestä **Friday** voidaan muodostaa käsite, joka tulkitaan ajanjakson perjantai lopettamien ajanjaksojen joukoksi. Yleisellä eksistentiaalisella kvantifoinnilla **during** ominaisuuden avulla käsitteestä **Week** voidaan muodostaa käsite, joka tulkitaan ajanjakson viikko aikana esiintyvien ajanjaksojen joukoksi. Muodostamalla leikkaus kaikkien näiden käsitteiden ja käsitteen **TimeInterval** välille muodostuu käsite, joka tulkitaan viikonaikana maanantaista perjantaihin kestävien ajanjaksojen joukoksi:



$$\begin{aligned}
\text{MaanantaistaPerjantaihin} &\equiv \text{TimeInterval} \\
&\sqcap (\exists \text{startsBy.Monday}) \\
&\sqcap (\exists \text{finishesBy.Friday}) \\
&\sqcap (\exists \text{during.Week}).
\end{aligned}$$

Päivähoito ei kuitenkaan ole yhtäjaksoisesti maanantaista perjantaihin kestävä prosessi, vaan tähän ajanjaksoon kuuluvina päivinä vain osan päivästä kestävä prosessi. Niinpä määrittelemme käsitteen ”osan päivästä” (*OsanPäivästä*). Käsitteelle **TimeInterval** on määritelty ominaisuus ajanosa (**temporalPart**). Sillä on mahdollista määritellä jonkin ajanjakson ajanosien muodostamia joukkoja. Yleisellä eksistentiaalisella kvantifioinnilla **temporalPart** ominaisuuden avulla käsitteestä **Day** voidaan muodostaa käsite, joka tulkitaan päivän ajanosien muodostamaksi joukoksi:

$$\text{OsanPäivästä} \equiv (\exists \text{temporalPart.Day}).$$

Muodostamalla leikkaus tämän ja käsitteen *MaanantaistaPerjantaihin* kanssa muodostuu käsite, joka tulkitaan maanantaista perjantaihin osanpäivästä kestävien ajanjaksojen joukoksi (*MaanantaistaPerjantaihinOsanPäivästä*):

$$\begin{aligned}
\text{MaanantaistaPerjantaihinOsanPäivästä} &\equiv \text{MaanantaistaPerjantaihin} \\
&\sqcap \text{OsanPäivästä}.
\end{aligned}$$

Käsite ”leikinohjaus tai lastenhoito” *LeikinOhjausTaiLastenhoito* määritellään yhdisteen avulla omaksi nimetyksi kokonaisuudeksi. Muodostamalla yhdiste käsitteiden *LeikinOhjaus* ja *Lastenhoito* välille muodostuu käsite, joka tulkitaan leikin ohjauksen tai lastenhoitojen muodostamaksi joukoksi:

$$\text{LeikinOhjausTaiLastenhoito} \equiv \text{LeikinOhjaus} \sqcup \text{Lastenhoito}.$$

Yleisellä eksistentiaalisella kvantifioinnilla **time** ominaisuuden avulla käsitteestä *MaanantaistaPerjantaihinOsanPäivästä* voidaan muodostaa käsite, joka tulkitaan maanantaista perjantaihin osan päivästä esiintyvien fyysikaalisten joukoksi. Muodostamalla leikkaus tämän ja käsitteen *LeikinOhjausTaiLastenhoito* välillä muodostuu käsite, joka tulkitaan maanantaista perjantaihin osan päivästä esiintyvien leikin ohjausten tai lastenhoitojen joukoksi:

$$\text{Leikkitoiminta} \equiv \text{LeikinOhjausTaiLastenhoito} \\ \sqcap (\exists \text{time.MaanantaistaPerjantaihinOsanPäivästä}).$$

Viimeisenä määritellään käsite `LastenKotihoidonTuki`. Sen määritelmä STAKES-Palveluluokituksessa on seuraava: ”kunnallisen päivähoidon vaihtoehtona maksettava sosiaaliavustus lapsen hoitamiseksi kotonaan. Sisältää myös mahdollisen lasten kotihoidon tuen kuntalisän”. Tämän määritelmän keskeisin tietosisältö on ”avustus lapsen hoitamiseksi kotonaan”, josta muodostetaan käsitteen `LastenKotihoidonTuki` formaali määritelmä.

Käsite `LastenKotihoidonTuki` jaetaan primitiivisemmiksi käsitteiksi: lasten kotihoito (`LastenKotihoiti`) ja tukeminen. Edellä on määritelty käsite `Lastenhoito`, josta voidaan helposti yleisellä eksistentiaalisella kvantifioinnilla **located** ominaisuuden avulla määritellä käsite `LastenKotihoito`:

$$\text{LastenKotihoito} \equiv \text{LastenHoito} \\ \sqcap (\exists \text{located.SingleFamilyResidence}).$$

SUMO-ontologiassa on edustettuna käsite rahoittaminen `Funding`, joka määritellään rahalliseksi tukemiseksi. Tämän käsitteen tulkinta kattaa rahan antamisen toiselle toimijalle, jonka ei odoteta antavan mitään takaisin ja rahan antamisen toiselle toimijalle sillä ehdolla, että rahan saaja antaa takaisin rahaa sen antajalle tietyn prosenttiosuuden verran saajan mahdollisista tulevista tuotoista.

Yleisellä eksistentiaalisella kvantifioinnilla **patient** ominaisuuden avulla käsitteestä `LastenKotihoito` voidaan muodostaa käsite, joka tulkitaan lasten kotihoitoon kohdistuvien prosessien joukoksi. Muodostamalla leikkaus tämän ja käsitteen `Funding` välille muodostuu käsite, joka tulkitaan lasten kotihoitoon kohdistuvien rahallisten tukemisien joukoksi:

$$\text{LastenKotihoidonTuki} \equiv \text{Funding} \\ \sqcap (\exists \text{patient.LastenKotihoito})$$

Lasten kotihoidontuen välittömänä vastaanottajana ovat lasten vanhemmat. Käsitettä `LastenKotihoidonTuki` voidaan tarkentaa määrittelemällä lasten vanhemmat

tuen välittömiksi saajiksi. Käsite **Funding** perii käsitteeltä **Process**, sen epäsuorana alakäsitteenä, ominaisuuden **destination**, jolla voidaan määritellä prosessille määränpää tai vastaanottaja. Yleisellä eksistentiaalisella kvantifiointilla **destination** ominaisuuden avulla käsitteestä lapsen vanhempi (**LapsenVanhempi**) voidaan muodostaa käsite, joka tulkitaan lapsen vanhemmat vastaanottajana omaavien prosessien joukoksi. Muodostamalla leikkaus tämän ja edellä määritellyn käsitteen välille muodostuu käsite, joka tulkitaan lasten vanhemmat vastaanottajana omaavien lasten kotihoitoon kohdistuvien rahallisten tukemisien joukoksi:

$$\begin{aligned} \text{LastenKotihoidonTuki} &\equiv \mathbf{Funding} \\ &\sqcap (\exists \text{patient}.\text{LastenKotihoito}) \\ &\sqcap (\exists \text{destination}.\text{LapsenVanhempi}). \end{aligned}$$

Käsite **LapsenVanhempi** voidaan määritellä ominaisuuden **parent** käänteisominaisuuden **parentOf** avulla seuraavasti

$$\begin{aligned} \text{LapsenVanhempi} &\equiv \mathbf{Human} \\ &\sqcap (\exists \text{parentOf}.\text{Lapsi}) \end{aligned}$$

eli yleisellä eksistentiaalisella kvantifiointilla **parentOf** ominaisuuden avulla käsitteestä **Lapsi** muodostetaan käsite, jonka käsitteen **Human** kanssa muodostama leikkaus tulkitaan lapsen omaavien ihmisten muodostamaksi joukoksi.

Tässä määritellyt käsitteet on esitetty XML-esitysmuotoon sarjallistettuna liitteessä 1.

## 7.5 Käsitteiden väliset automaattisesti pääteltävät suhteet

STAKES-Palveluluokituksessa kuvailluista käsitteistä edellä määriteltiin OWL+SUMO-menetelmällä kuusi käsitettä: **Ensihoito**, **Sairaankuljetus**, **Leikkaustoiminta-Lastenkirurgia**, **Lastenneuvolapalvelu**, **Leikkitoiminta** ja **LastenKotihoidonTuki**. Näihin viitataan nimellä STAKES-esimerkkikäsitteet. Lisäksi edellä määriteltiin samalla menetelmällä useita STAKES-esimerkkikäsitteiden merkitykseen sisältyviä primitiivisempiä käsitteitä. Nämä ja STAKES-esimerkkikäsitteet muodostavat yhdessä noin 40:n käsitteen käsitejärjestelmän, johon viitataan symbolilla  $\mathcal{T}$ . Seuraavaksi tarkastellaan käsitejärjestelmän  $\mathcal{T}$  sisältämien ja yleisesti

OWL+SUMO-menetelmällä määriteltyjen käsitteiden välisiä implisiittisiä sisältymis- ja vastaavuussuhteita, jotka on mahdollista löytää automaattisesti päättelykoneiden avulla. Suhteiden päättelyssä käytetään Pellet-päättelykonetta.

Käsitteitä voidaan määritellä ja käyttää päättelykoneiden avulla kyselyjen tekemiseen. Oletetaan, että toimijan  $t_1$  tarjoamat palvelut on kuvailtu käsitejärjestelmällä  $\mathcal{T}$ . Oletetaan lisäksi, että englanninkielinen toimija  $t_2$  tarvitsee yleisesti lapsiin kohdistuvia palveluita. Toimija  $t_2$  voi selvittää tarjoaako toimija  $t_1$  tällaisia palveluita muodostamalla OWL+SUMO-menetelmällä käsitteen ”lapsiin kohdistuvat” (engl. child targeted) (**Child-targeted**) ja kysymällä päättelykoneelta tämän käsitteen tulkintaan sisältyviä käsitteitä käsitejärjestelmässä  $\mathcal{T}$ .

Käsitteen **Child-targeted** määrittelemiseksi toimijan  $t_2$  on muodostettava määritelmä käsitteelle lapsi (**Child**). Toimija  $t_2$  määrittelee lapsen alle 18-vuotiaaksi ihmiseksi:

$$\begin{aligned} \mathbf{Child} &\equiv \mathbf{Human} \\ &\sqcap \leq \mathbf{age} \\ &\sqcap (\exists \mathbf{age}.\mathbf{Under18Years}). \end{aligned}$$

Käsitteen ”alle 18 vuotta” (**Under18Years**) hän määrittelee seuraavasti:

$$\begin{aligned} \mathbf{Under18Years} &\equiv \mathbf{YearDuration} \\ &\sqcap (\exists \mathbf{MagnitudeFn}.\mathbf{lessThan18}). \end{aligned}$$

Määritelmässä esiintyvän välille nolasta kahdeksaantoista kuuluvien positiivisten kokonaislukujen joukoksi tulkittavan tietotyypin **lessThan18** toimija  $t_2$  määrittelee XML-SCHEMA2-suosituksella seuraavasti:

```
<simpleType name=''lessThan18''>
  <restriction base=''positiveInteger''>
    <maxInclusive value=''17''/>
  </restriction>
</simpleType>.
```

Nyt toimija  $t_2$  voi muodostaa yleisellä eksistentiaalisella kvantifioinnilla **patient** ominaisuuden avulla käsitteestä **Child** käsitteen, joka tulkitaan kohteenaan alle 18-vuotiaiden ihmisten muodostamaan joukkoon kohdistuviksi entiteeteiksi:

$$\text{Child-targeted} \equiv \exists \text{patient.Child.}$$

Käsitejärjestelmässä  $\mathcal{T}$  käsitteiden **Lastenhoito**, **AlleKouluikäisenRokotuspalvelu**, **AlleKouluikäisenTutkimuspalvelu**, **AlleKouluikäisenTerveysneuvonta** ja **Leikkaustoiminta-Lastenkirurgia** määritelmät päätellään sisältyväksi käsitteen **Child-targeted** merkitykseen. Toisin sanoen ne kuvailevat toimijan  $t_1$  tarjoamia palveluita, jotka kohdistuvat toimijan  $t_2$  määritelmän mukaisiin lapsiin. Jos STAKES-Palveluluokituksen sisältämät käsitteet on määritelty OWL+SUMO-menetelmällä, niin siitä löydetään automaattisesti kyselykäsitteeseen esimerkiksi käsitteeseen lapsiin kohdistuvat palvelut sisältyvät käsitteet.

Toimija  $t_2$  voi myös tarkistaa kuinka hänen määrittelemänsä käsite **Child** suhteutuu toimijan  $t_1$  määrittelemiin käsitteisiin. Toimija  $t_2$  voi esimerkiksi päättelijän avulla etsiä hänen määritelmänsä vastaavia käsitteitä. Ilmenee, että vastaavia käsitteitä ei käsitejärjestelmässä  $\mathcal{T}$  ole. Käsitejärjestelmän käsite **Lapsi** päätellään sisältyväksi käsitteen **Child** merkitykseen. Toimijan  $t_1$  käsitys lapsesta on siis suppeampi kuin toimijan  $t_2$  käsitys. Puhtaasti syntaksiin perustuvassa vastaavassa haussa toimijalta  $t_2$  olisi voinut jäädä saamatta hänen mielestään relevantteja toimijan  $t_1$  tarjoamia palveluita. OWL+SUMO-menetelmällä päättelykoneen avulla toteutetussa haussa toimija  $t_2$  saa vastauksenaan täsmällisesti haluamansa palvelut, riippumatta toimijan  $t_1$  tulkinnasta toimijan  $t_2$  käyttämälle termille.

STAKES-palveluluokituksesta ei ole käsitettä ”lapsiin kohdistuva”, jonka avulla löytyisivät kaikki lapsiin kohdistuvat palvelut. Jos käsitteistön käsitteet on kuvailtu ainoastaan luonnollisella kielellä, niin jonkin mielivaltaisen käsitteen ekstensioon kuuluvat entiteetit pyritään tyyppillisesti löytämään automaattisesti vapaalla tekstihaulla. Haettaessa vapaalla tekstihaulla lapsiin kohdistuvia palveluita, vastauksena saadaan tämän hakutyypin ongelmista johtuen myös käsite ”Lasten kotihoidon tuki”, joka ei ole välittömästi lapsiin kohdistuva vaan heidän vanhempiansa kohdistuva palvelu. Tätä on kuitenkin mahdotonta tulkita puhtaasti syntaksiin perustuvilla menetelmillä, koska kyseisen käsitteen määritelmässä esiintyy termi lapsi ja vieläpä

tekemisen kohteena:”kunnallisen päivähoidon vaihtoehtona maksettava sosiaaliavustus lapsen hoitamiseksi kotonaan. Sisältää myös mahdollisen lasten kotihoidon tuen kuntalisän”.

Oletetaan, että toimija  $t_2$  etsii STAKES-esimerkkikäsitteistä SUMO-ontologiassa määritellyn käsitteen **TherapeuticProcess** avulla hoitopalveluita tarkemmin kuvaavia käsitteitä. Näistä käsitteen **TherapeuticProcess** alakäsitteiksi tulkitaan **Ensihoito** ja **Leikkaustoiminta-Lastenkirurgia**. Oletetaan nyt, että toimija  $t_2$  etsii STAKES-palveluluokituksen STAKES-esimerkkikäsitteitä vastaavista käsitteistä vapaalla tekstihaulla hoitopalveluita. Esimerkiksi hakusanalla ”hoito” ei löydetä käsitettä **Leikkaustoiminta-Lastenkirurgia**, koska sen nimessä ei esiinny sanaa ”hoito” eikä siihen liity määritelmää, jossa tämä sana esiintyisi. Hakusanalla ”hoito” löydetään käsite **Ensihoito**, mutta lisäksi löydettäisiin todennäköisesti käsite **Leikkitoiminta**, koska sen määritelmässä esiintyy sana ”päivähoito”. **Leikkitoiminta** ei kuitenkaan ole toivottu hakutulos, jos sanalla hoito viitataan parantaviin hoitoihin.

Luonnollisessa kielessä eri merkityksiin voidaan viitata samalla termillä kuten edellä parantavaan hoitoon ja huolehtimiseen viitataan samalla termillä ”hoitaminen”. Termin merkitys voi kontekstista riippuen vaihdella, joka aiheuttaa ongelmia syntaksiin perustuville menetelmille. OWL+SUMO-menetelmällä päättelykoneen avulla tehdyssä kyselyssä käsitettä **Leikkitoiminta** käsitettä ei saada vastauksena, koska kyselykäsitteenä käytettiin nimenomaan parantavien hoitojen joukoksi tulkittavaa primitiivikäsitettä **TherapeuticProcess**.

Kuvitteellisen toimijan  $t_2$  edellä määrittelemä kyselykäsite ja sen merkitykseen sisältyvät primitiivisemmät käsitteet on esitetty XML-esitysmuotoon sarjallistettuna liitteessä 2.

## 8 Kuvauslogiikka+primitiivikäsitteistö-menetelmän arviointi

Tämän tutkielman tavoitteena oli esitellä menetelmä, jolla toimijat voivat kuvailla tiedon merkityksen webissä niin, että tämä tieto on automaattisesti tunnistettavissa eli suhteutettavissa muuhun samalla menetelmällä webissä kuvailtuun tietoon. Tutkielmassa esiteltiin kuvauslogiikka+primitiivikäsitteistö-menetelmä, jolla tiedon merkitys voidaan kuvailla logiikkaperusteisesti määritellyillä käsitteillä.

Kuvauslogiikka+primitiivikäsitteistö menetelmän tarkastelemiseksi ja testaamiseksi sitä sovellettiin muutamaaan STAKES-palveluluokituksen sisältämään käsitteeseen (STAKES-esimerkkikäsitteeseen). Menetelmästä sovellettiin OWL+SUMO-ilmentymää eli kuvauslogiikkana käytettiin OWL-kuvauslogiikkaa ja primitiivikäsitteistönä SUMO-ontologiaa. Osoittautui, että tällä menetelmällä on mahdollista määritellä käsitteitä kuvailemaan erilaisia kohteita, esimerkiksi tilastointia tai yleisemmin tiedonhakua varten. Lisäksi osoittautui, että jos toimijat määrittelevät tarvitsemansa käsitteet tällä menetelmällä, niin eri toimijoiden määrittelemien käsitteiden semanttiset sisältymissuhteet voidaan löytää automaattisesti.

### 8.1 OWL+SUMO-menetelmän ilmaisuvoima

STAKES-esimerkkikäsitteiden tarkentavina piirteinä toimivat pääosin palvelun kohde ja sen toteuttaja. Esimerkiksi palvelukäsitteen sairaankuljetus toteuttajana on hoitoalan ammattihenkilö ja kohteena sairas ihminen. Joitakin kertoja tarkentavana piirteenä toimi palvelun sijaintipaikka tai toteutusajankohta. Palvelun tarkentavana piirteenä esiintyi myös muiden muassa palvelun päämäärä.

Koska kuvailtaviksi käsitteiksi oli valittu lapsiin ikäryhmän edustajina kohdistuvia palveluita, niin palvelun kohdetta tarkentavana piirteenä oli pääsääntöisesti kohteen ikä. Palvelun toteuttajaa tarkennettiin pääosin kuvailemalla prosessi, jonka palvelun toteuttaja taitaa eli prosessi jonka suhteen palvelun toteuttaja on ammattitaitoinen. Palvelun toteutusajankohtaa tarkennettiin kuvailemalla niiden ajallista suhdetta johonkin prosessiin tai toiseen ajankohtaan.

STAKES-esimerkkikäsitteiden luonnollisen kielen määritelmät pystyttiin formalisoimaan pääosin suoraan SUMO-ontologian avulla. Muutamissa tapauksissa tämä ei ollut mahdollista. Tällöin STAKES-esimerkkikäsitteelle jouduttiin tekemään sen luonnollisen kielen määritelmästä merkitykseltään hieman poikkeava tai puutteellinen määritelmä, kuten käsitteelle palvelu. Arviolta noin puolet STAKES-palveluluokituksessa esitellyistä luonnollisen kielen määritelmistä sisälsivät käsitteitä, joiden määrittely on todennäköisesti ongelmalliseksi SUMO-ontologian käsitteiden ja ominaisuuksien avulla. Arviolta noin puolet STAKES-palveluluokituksessa esitellyistä luonnollisen kielen määritelmistä on mahdollista formalisoida suoraan SUMO-ontologian avulla. Arviolta reilusti yli puolelle STAKES-palveluluokituksessa esitellyistä luonnollisen kielen määritelmistä on mahdollista muodostaa niitä semantiikkaltaan pitkälti vastaava saman tarkkuustason formaali määritelmä SUMO-ontologian avulla. Vaikka SUMO-ontologian lehtitason käsitteet ovat paikoin liian yleisiä joidenkin perustason käsitteiden täsmällisten määritelmien muodostamiseksi, niin kokonaisuutena se soveltuu hyvin eri sovellusalojen perustason tai niitä hieman erityisempien käsitteiden kuvailuun osana kuvauslogiikka+primitiivikäsitteistö menetelmää.

STAKES-esimerkkikäsitteiden määrittelyyn tarvittava ilmaisuvoiman taso oli  $\mathcal{ALC}HN\mathcal{I}(\mathcal{D})$  eli niiden määrittelyyn olisi siis riittänyt ilmaisuvoimaltaan OWL-kuvauslogiikkaa hieman köyhempikin kieli. Määrittelyissä selvästi käytetyin konstruktori oli täydellinen eksistentiaalinen kvantifiointi. Seuraavaksi käytetyin oli leikkaus. Jokaisessa käsittemäärittelyssä käsitteen nimi liitettiin sen formaaliin määritelmään vastaavuus- tai alakäsiteaksiomalla. Käänteisominaisuuden määrittelevää aksiomaa tarvittiin myös useasti. Yhdisteen, numerorajoitteen ja tietotyypin määrittelylle syntyi myös tarvetta, mutta selkeästi edellä mainittuja määrittelyitä vähemmän. OWL-kuvauslogiikan ilmaisuvoiman taso vaikuttaa sopivalta erilaisten sovellusalojen käsittemäärittelyjen muodostamiseen.



## 8.2 Kuvauslogiikka+primitiivikäsitteistö-menetelmän hyötyjä

Kuvauslogiikka+primitiivikäsitteistö-menetelmän avulla toimijat voivat määrittellä tarvitsemiaan käsitteitä toisistaan riippumatta ja löytää kaikkien tällä menetelmällä määriteltyjen käsitteiden väliset semanttiset sisältymissuhteet automaattisesti. Semanttisen yhteentoimivuuden mahdollistamisen lisäksi tällä menetelmällä voidaan helpottaa myös käsitteistöjen kehitystä ja ylläpitoa.

Nykyisin tietojärjestelmissä tiedon merkityksen kuvailuun käytettyjen käsitteistöjen hierarkia muodostetaan ja ylläpidetään pääosin ihmisvoimin. Tämä on virhealtista ja työlästä. Muutokset hierarkian yhdessä osassa voivat edellyttää hierarkian muihin osiin muutoksia, jotka jäävät ihmiseltä helposti huomaamatta ja voivat olla työläistä tehdä. Esimerkiksi jos käsitteistöön lisätään uusi käsite, niin useat käsitteistöissä jo olleet käsitteet voivat olla merkitykseltään päällekkäisiä tämän uuden käsitteen kanssa. Näiden päällekkäisyyksien eksplisiittinen määrittely voi olla työlästä ja ihmiseltä saattaa helposti jäädä huomaamatta osa näistä päällekkäisyyksistä. Muutokset käsitteistöissä, joiden hierarkiaa ylläpidetään ihmisvoimin ovat tyypillisesti ajan suhteen vähäisiä ja siksi tällaisiin käsitteistöihin viitataan tässä nimellä ”staattinen käsitteistö”.

Joustavan ja yleiskäyttöisen staattisen käsitteistön kehittämiseen liittyy muitakin haasteita. Eri toimijat ja käyttötavat asettavat kukin runsaasti vaatimuksia, joiden kaikkien täyttäminen staattisella käsitteistöllä on erittäin vaikeata [Oja05b].

Yleiskäyttöinen staattinen käsitteistö voidaan pyrkiä toteuttamaan kuten STAKES-palveluluokitus eli listaamalla kaikki sovellusalueella käytössä olevat käsitteet ja määrittelemällä niille yläkäsitteet, joiden alle käytössä olevat listatut käsitteet jäsennetään [Oja05b]. Tällainen ”alhaalta ylös” (engl. bottom up) -menetelmä soveltuu käytettäväksi hyvin silloin, kun kehitetään luomishetkellä tunnettuja olemassa olevia tarpeita jäsentäviä käsitteistöjä. Alahaalta ylös -menetelmällä syntyy tyypillisesti rakenteeltaan suhteellisen sovelluskeskeinen käsitteistö, joka on käytännössä toimiva ja tehokas niin kauan kun siihen kohdistuvat tarpeet eivät poikkea sen luomishetkellä tunnetuista tarpeista.

Sovelluskeskeisen staattisen käsitteistön muokkaaminen uusia tarpeita vastaavaksi voi olla hankalaa. Uusi tarve ei välttämättä jäsenny mielekkäästi rakenteeseen, joka jäsentää ainoastaan aiemmin tunnettuja olemassa olevia tarpeita. Niinpä uusia tarpeita vastaavien muutosten tekeminen sovelluskeskeiseen staattiseen käsitteistöön saattaa vaatia sen rakenteeseen laajasti heijastuvia muutoksia. Sovelluskeskeisen staattisen käsitteistön ongelmana voi siis monesti olla se, että se vastaa huonosti tulevaisuuden tarpeisiin ja sen ylläpitäminen on hankalaa.

Alhaalta ylös -menetelmällä kehitetyissä käsitteistöissä esiintyviä tyypillisiä ongelmia voidaan välttää kehittämällä käsitteistö ylhäältä alas (engl. top down) -menetelmällä. Tässä menetelmässä kartoitetaan aluksi käsitteistön sovellusalueeseen liittyvät yleiset käsitteet ja niiden mahdolliset tarkentavat piirteet. Tämän jälkeen yleisille käsitteille määritellään alakäsitteitä systemaattisesti lisäämällä niiden määritelmään tarkentavia piirteitä ja näiden muodostamia yhdistelmiä.

Ylhäältä alas -menetelmällä syntyy luontevasti alhaalta ylös -menetelmään nähden rakenteeltaan sovellusriippumattomia eli yleisiä käsitteistöjä. Rakenteeltaan yleinen staattinen käsitteistö tukee todennäköisesti paremmin tulevaisuuden muuttuvia tarpeita, kuin rakenteeltaan sovelluskeskeinen staattinen käsitteistö. Sitä on myös todennäköisesti helpompi ylläpitää, koska ylhäältä alas -menetelmällä muodostetun yleisen staattisen käsitteistön tavoitteena on nimenomaan lähtökohtaisesti huomioida systemaattisesti kaikki mahdolliset relevantit tarpeet. Yleisten käsitteiden systemaattinen tarkentaminen toimialakohtaisiksi käsitteiksi johtaa kuitenkin helposti tai lähes väistämättä kombinatoriseen räjähdykseen, jossa yhdenmukaisuuden säilyttämiseksi käsitteille muodostetaan tarkentavien piirteiden ja näiden muodostamien yhdistelmien avulla alakäsitteitä eksponentiaalisesti kasvavissa määrin [Oja05b]. Rakenteeltaan yleisen staattisen käsitteistön suunnittelu ja rajaaminen sovellusalueen tarpeita jäsentäväksi on erittäin vaativaa ja työlästä. Tällaisen käsitteistön kehittämisessä uhrautuu helposti paljon työtä käsitelmäärittelyihin, joita ei koskaan käytännössä hyödynnetä [Oja05b].

Kuvauslogiikka+primitiivikäsitteistö-menetelmällä määriteltyjen käsitteiden väliset semanttiset sisältymissuhteet on mahdollista löytää automaattisesti, olemassa olevilla standardipäättelyn toteuttavilla päättelykoneilla. Tästä seuraa se, että tällä

menetelmällä määritellyt käsitteet voidaan jäsentää (luokitella) automaattisesti hierarkiseksi käsitteistöksi. Näin toimijat voivat keskittyä määrittelemään ainoastaan olemassa olevia tarpeita vastaavia käsitteitä eikä heidän tarvitse pohtia systemaattisesti rakentuvaa yleistä rakennetta, jota voidaan suoraviivaisesti laajentaa vastaamaan tulevaisuuden tarpeita. Tarvittavat muutokset voidaan tehdä helposti niitä tarvittaessa, koska käsitteiden lisäämisestä, muokkaamisesta ja poistamisesta seuraavat muutokset käsitteistön rakenteeseen voidaan tuottaa automaattisesti.

Staatinen käsitteistö tukee sillä kuvailtujen kohteiden  $O$  hakua tai hyödyntämistä ainoastaan siinä luetelluilla käsitteillä. Tiedonhaussa tai kohteiden hyödyntämisessä voi kuitenkin syntyä tarve käsitteelle, jota staatteisesta käsitteistöstä ei löydy, mutta joka on sen kanssa semanttisesti päällekkäinen. Jos kohteiden hyödyntämistä varten olevat käsitteet on määritelty kuvauslogiikka+primitiivikäsitteistömenetelmällä, niin tiedonhakijaa voi määrittellä tarvitsemansa käsitteet kuvauslogiikka+primitiivikäsitteistömenetelmällä ja pyytää päättelykonetta etsimään automaattisesti hänen määrittelemiensä käsitteiden merkitykseen sisältyvät kohteiden  $O$  hyödyntämistä varten alun perin tarkoitetut käsitteet.

Jos esimerkiksi STAKES-palveluluokituksen sisältämät käsitteet olisi määriteltyinä kuvauslogiikka+primitiivikäsitteistömenetelmällä, niin eri toimijat (esimerkiksi tilastojat, tiedonhakijat tai soster-huolto palveluita tarjoavat organisaatiot) voisivat määrittellä vapaasti tarvitsemiaan käsitteitä samalla menetelmällä ja löytää automaattisesti STAKES-palveluluokituksen ja määrittelemiensä käsitteiden väliset semanttiset sisältymissuhteet.

### 8.3 OWL+primitiivikäsitteistö-menetelmän soveltamishaasteita

Klassinen teoria väittää, että kohteet joko kuuluvat tai eivät kuulu johonkin tiettyyn kategoriaan. Ihmisille ei kuitenkaan monesti ole täysin selvää mihin kategoriaan jokin yksilö kuuluu. Ihmisen muistissa kategoriaan kuulumiseen liittyy epävarmuutta ja riippumattomuutta yksittäisistä piirteistä. Klassiseen teoriaan perustuvalla kuvauslogiikalla tällaista käsitteisiin ihmisen muistissa liittyvää sumeutta ei ole mahdollista

kuvailla. Niinpä kuvauslogiikalla voi monissa tapauksissa olla hyvin vaikeata määrittellä käsitteitä kategoriaan kuulumiselle vaadittavina täsmällisinä välttämättöminä ja riittävinä ehtoina.

Käsitteiden tarkoituksenmukainen määrittely kuvauslogiikka+primitiivikäsitteistömenetelmällä vaatii kuvauslogiikan hallitsemisen lisäksi käytettävän primitiivikäsitteistö-  
menetelmällä vaatii kuvauslogiikan hallitsemisen lisäksi käytettävän primitiivikäsitteistö-  
menetelmällä vaatii jo menetelmän tasolla erityisosaamista, jota käsitteiden määrittely vapail-  
la tekstikuvauksilla ei juurikaan vaadi. Esimerkiksi juuri tämän, menetelmän vaa-  
tivuuden, johdosta tämän tutkielman puitteissa ei ollut mahdollista testata sitä,  
kuinka välittömästi eri ihmisten SUMO-ontologian avulla tuottamat määritelmät  
samoista käsitteistä sisältyvät toisiinsa. Jos primitiivikäsitteistö tarjoaa mahdolli-  
suuden määrittellä käsitteen useilla eri primitiivikäsitteillä ja -ominaisuuksilla, niin  
on mahdollista, että eri ihmiset määrittelevät merkitykseltään vastaavat käsitteet  
tulkintafunktiolla tulkittuna merkitykseltään etäisiksi.

OWL-kuvauslogiikan ilmaisuvoimaa on rajoitettu, jotta sillä esitetyssä tietämyk-  
sessä mahdollisesti esiintyvät päättelyongelmat ovat ratkeavia [GHVD03]. Niinpä  
OWL-kuvauslogiikalla on mahdollista määrittellä sääntöjä vain hyvin rajoitetusti  
verrattuna esimerkiksi predikaattilogiikkaan. OWL-kuvauslogiikka on predikaattilo-  
giikan ratkeava osajoukko. Niinpä predikaattilogiikan avulla on mahdollista havain-  
nollistaa OWL-kuvauslogiikan ilmaisuvoiman rajoittuneisuutta.

Kuvauslogiikan käsitettä  $C$  vastaa predikaattilogiikassa yksipaikkainen predikaatti  
 $C(x)$  ja kuvauslogiikan ominaisuutta  $R$  vastaa predikaattilogiikassa kaksipaikkai-  
nen predikaatti  $P(x, y)$ , joissa  $x$  ja  $y$  ovat muuttujia. OWL-kuvauslogiikka vastaa  
oleellisesti predikaattilogiikkaa, jossa on sallittu ainoastaan yksi- ja kaksipaikkai-  
sten predikaattien käyttö. Lisäksi kaksipaikkaisessa predikaatissa eksistentiaalisesti  
tai universaalisti kvantifioidun muuttujan on aina esiinnyttävä vapaan muuttujan  
kanssa [GHVD03]. Esimerkiksi kuvauslogiikan yleistä eksistentiaalista kvantifiointia  
 $\exists P.C$  vastaa predikaattilogiikassa seuraava kaava [GHVD03]:

$$\exists y.(P(x, y) \wedge C(y)).$$

OWL-kuvauslogiikassa yleisellä eksistentiaalisella kvantifioinnilla suurempi kuin omi-

naisuuden avulla käsitteestä **Planeetta** voidaan muodostaa esimerkiksi käsite, joka tulkitaan vähintään yhtä planeettaa suurempien entiteettien joukoksi:

$$\exists \text{suurempiKuin.Planeetta.}$$

Vastaavasti universaalilla kvantifioinnilla **suurempiKuin** ominaisuuden avulla käsitteestä **Planeetta** voidaan muodostaa käsite, joka tulkitaan ainoastaan planeettoja suurempien entiteettien joukoksi:

$$\forall \text{suurempiKuin.Planeetta.}$$

Näitä määritelmiä on mahdollista yhdistellä, mutta OWL-kuvauslogiikassa ei ole mahdollista määritellä esimerkiksi sääntöä ”kaikki tähdet ovat suurempia kuin kaikki planeetat”. Predikaattilogiikan avulla tämä on helppo ilmaista:

$$(\forall x)(\forall y)\text{Tähti}(x) \wedge \text{Planeetta}(y) \Rightarrow \text{suurempiKuin}(x, y).$$

Käsitteiden kuvauslogiikkaperustaisessa määrittelyssä ja yleisestikin konetulkittavan semantiikan formaalissa määrittelyssä yksi helposti unohtuva asia on lisätä käsitteiden määritelmään taustatietämystä. Tällainen tietämys on tyypillisesti piilevää ja ihmiselle niin itsestään selvää, että sellaisen eksplisiittinen määrittely usein unohtuu. **HoitopaikkaanSiirtyminen** ja **Sairaankuljetus** ovat esimerkiksi hoitopaikkaan siirtymisen aikaisia (**HoitopaikkaanSiirtymisenAikainen**) prosesseja, jota päättelijä ei voi edellisessä luvussa määriteltyjen käsitteiden perusteella päätellä. Toisin sanoen jos käsitteen **HoitopaikkaanSiirtymisenAikainen** määritelmään ei lisätä leikkausta käsitteiden **Transfer** ja **HoitopaikkaanSiirtymisenAikainen** välille, niin sen määritelmästä ei voi päätellä, että kyseessä on hoitopaikkaan siirtymisen aikainen prosessi. Sama pätee käsitteeseen **Sairaankuljetus**.

Taustatietämyksen määrittely yhdistää käsitteitä ja parantaa siten oleellisesti semanttisesti läheisten käsitteiden löytymistä. Jos käsitteiden **HoitopaikkaanSiirtyminen** ja **Sairaankuljetus** määritelmiin lisätään leikkauksella käsite **HoitopaikkaanSiirtymisenAikainen**, niin niille saadaan seuraavat määritelmät:

**HoitopaikkaanSiirtyminen**  $\equiv$  **Translocation**

$\sqcap$  ( $\exists$ **destination**.Hoitopaikka).

$\sqcap$  **HoitopaikkaanSiirtymisenAikainen**

ja

**Sairaankuljetus**  $\equiv$  **Transfer**

$\sqcap$  ( $\exists$ **agent**.HoitoalanAmmattihenkilö)

$\sqcap$  ( $\exists$ **patient**.Sairas)

$\sqcap$  ( $\exists$ **destination**.Hoitopaikka)

$\sqcap$  **HoitopaikkaanSiirtymisenAikainen**.

Taustatietämyksestä on erityisesti apua tilanteissa, joissa yritetään etsiä johonkin käsitteeseen läheisesti, mutta sen tarkemmin määrittelemättä liittyvät käsitteet. Tällaisia käsitteitä ovat tyypillisimmin käsitteen sisarkäsitteet ja näiden sisarkäsitteiden alakäsitteet. Käsitteen  $K$  sisarkäsitteitä ovat kaikki käsitteen  $K$  yläkäsitteen muut alakäsitteet.

Oletetaan esimerkiksi että halutaan löytää STAKES-esimerkkikäsitteiden joukosta kaikki käsitteeseen **Ensihoito** läheisesti liittyvät käsitteet. Käsitteen **Ensihoito** tulkinta päätellään sisältyväksi käsitteiden **Ammattihoito** ja **HoitoonSiirtymisenAikainenTaiTapaturmapaikallaOleva** tulkintaan. Käsitejärjestelmän  $\mathcal{T}$  käsitteistä näiden käsitteiden tulkintaan sisältyvät käsitteen **Ensihoito** lisäksi käsitteiden **TapaturmapaikallaOleva** ja **HoitopaikkaanSiirtymisenAikainen** tulkinat. Näiden käsitteiden tulkintaan STAKES-esimerkkikäsitteistä sisältyy käsitteen **Sairaankuljetus** tulkinta. Käsite **Sairaankuljetus** on käsitteen **HoitopaikkaanSiirtymisenAikainen** alakäsite. Ilman edellä määriteltyä taustatietämystä käsitettä **Sairaankuljetus** ei tunnistettaisi automaattisesti käsitteeseen **Ensihoito** läheisesti liittyväksi.

Vaikka OWL-kuvauslogiikalla esitetystä tietämyksestä mahdollisesti esiintyvät päätelyongelmat ovat ratkeavia, niin tällä esitetyn tietämyksen luokittelun pahimman tapauksen (engl. worst-case) aikavaativuus nykyisillä päättelijöillä on hyvin suuri, epädeterministinen eksponentiaalinen aikavaativuus [GHT06, SPG<sup>+</sup>05, HPSvH03].

Suuri aikavaativuus tulee esiin esimerkiksi jos nykyisillä päättelijöillä yritetään luokitella käsitteistö, jossa on määriteltynä kymmeniä tuhansia koosteisia käsitteitä [SR06]. Tämä ongelma on mahdollista kiertää, jos suuri käsitteistö pystytään jakamaan osiin [SR06].

## 9 Vastaavia menetelmiä

Terveydenhuollossa potilaan hoitoon liittyvät toimenpiteet suoritetaan monesti hajautetusti eri toimijoilla ja mahdollisesti eri organisaatioissa. Potilaalle myöhemmin tehtävät toimenpiteet ovat tyypillisesti riippuvaisia aiemmin tehdyistä toimenpiteistä ja niihin liittyvästä tietämyksestä. Tällaisten palveluketjujen automatisointi vaatii toimijoiden käyttämien tietojärjestelmien semanttista yhteentoimivuutta. Niinpä erityisesti terveydenhuollon toimialueen tutkimusprojekteissa on kehitelty erilaisia käsiteperustaisia tietämyksen kuvailumenetelmiä [Wro06], joista GALEN-projektissa<sup>19</sup> käytetty kuvailumenetelmä vastaa OWL+SUMO-menetelmää.

### 9.1 GALEN-projekti

GALEN-projektissa kehitettiin menetelmiä ja sovelluksia yhteiskäyttöisen ja uudelleen käytettävän lääketieteellisen tietämyksen määrittelyyn. Projektissa kehitettiin muiden muassa kuvauslogiikkaperustainen käsitteiden määrittelykieli GRAIL (GRAIL-kuvauskieli), jolla on mahdollista määrittellä OWL+SUMO-menetelmän tavoin primitiivikäsitteistä koosteisia käsitteitä. GALEN-projektissa määriteltiin the GALEN Common Reference Model (GALEN-käsitteistö) -niminen lääketieteen alan käsitteistö, jonka sisältämistä käsitteistä noin 10 000 määriteltiin primitiivikäsitteiksi<sup>20</sup>. Loput GALEN-käsitteistön 35 000:sta käsitteestä on GRAIL-kuvauskielen avulla primitiivikäsitteistä määriteltyjä koosteisia käsitteitä [RR05, Rec02].

GRAIL-kuvauskieli ja erityisesti GALEN-käsitteistön sisältämät primitiivit on suurimmaksi osaksi kehitetty lääketieteenalan varsin erityisten käsitteiden kuvailuun.

<sup>19</sup>[http://www.openclinical.org/prj\\_galen.html](http://www.openclinical.org/prj_galen.html)

<sup>20</sup>GALEN-projektia käsittelevissä artikkeleissa primitiivikäsitteisiin viitataan englanniksi nimellä elementary.

Niinpä GRAIL-kuvauskielen ja GALEN-käsitteistön muodostama menetelmä ei sellaisenaan sovellu yleiseksi toimialariippumattomaksi käsitteiden kuvailumenetelmäksi. GRAIL-kuvauskieli ei myöskään ole kokonaisuutena yhtä ilmaisuvoimainen kuin OWL-kuvauskieli. GRAIL-kuvauskielessä ei ole konstruktoria negaation ja yhdisteen määrittelyyn. Lisäksi joillakin sovellusalueilla ongelmalliseksi voi osoittautua se, että GRAIL-kuvauskielellä määrittelyssä tietämyksessä mahdollisesti esiintyvät päättelyongelmat voivat olla epätäydellisiä (engl. incomplete) [RR05].

## 9.2 ISO 15926 -standardi

Terveydenhuollon tavoin teollisuustuotannossa prosesseja hajautetaan useiden toimijoiden kesken. Teollisuuden tuottamiin tuotteisiin kertyy paljon tietoa koko niiden elinkaaren ajan. Tiedon tuottajina voivat toimia useat eri toimijat ja useiden eri toimijoiden on pystyttävä hyödyntämään tätä hajautetusti tuotettua tietoa tuotteen elinkaaren aikana [Wes04]. Tämän ongelman ratkaisemiseksi on kehitetty kansainvälinen standardi, jonka määrittelemällä ISO 15926 -menetelmällä voidaan kuvaila erityisesti prosessiteollisuudessa esiintyviin olioihin ja niiden elinkaariin liittyvää tietämystä [ISO03].

ISO 15926 -menetelmä jakautuu osiin, joista ISO 15926-2 -osaa [ISO03] voidaan pitää SUMO-ontologian tyyppisenä yläontologiana. ISO 15926-2 -yläontologia sisältää noin 200 primitiivistä käsitettä ja tietomallin, joiden avulla on mahdollista määrittellä erityisempiä käsitteitä [BWL<sup>+</sup>05]. ISO 15926 -menetelmän osa 15926-4 [ISO05a] on 15926-2 -yläontologian avulla määritelty prosessiteollisuuden ydinkäsitteistö. Se sisältää valmiiksi määriteltyjä prosessiteollisuusalan keskeisempiä käsitteitä, joita voidaan käyttää sellaisenaan tai joista voidaan edelleen määrittellä erityisempiä käsitteitä 15926-2 -yläontologian avulla. ISO 15926 -menetelmän osa ISO-15926-7 [ISO05b] määrittelee erilaisia toteutusmalleja muiden muassa syntaktisen yhteentoimivuuden toteuttamiseksi. ISO-15926-7 määrittelyn avulla voidaan esimerkiksi sarjallistaa ISO 15926 -menetelmällä kuvailtavaa tietämystä XML- tai OWL-esitysmuotoon [BWL<sup>+</sup>05].

ISO 15926 -menetelmällä määritellyillä käsitteillä tai luokilla on logiikkaperustainen semantiikka. Menetelmässä käytetty logiikkaa on kuitenkin varsin epätavallinen ja



vaikeasti ymmärrettävä joukko-oppi<sup>21</sup> [Smi06]. Ominaisuuksiin ja niiden mahdollisiin arvoihin, transitiivisuuteen, symmetriaan ja niin edelleen liittyvää teoriaa on myös arvosteltu erittäin epätavalliseksi ja vaikeasti ymmärrettäväksi [Smi06].

---

<sup>21</sup>ISO 15926 -menetelmässä käytettävän logiikan nimi on englanniksi non-well-founded-sets.

## 10 Yhteenveto

Tietojärjestelmien välisen yhteentoimivuuden toteuttaminen on ollut eräs suurimmista haasteista jo yli vuosikymmenen. Erityisesti näin on ollut yritystenvälisessä kaupankäynnissä ja muissa yritysten toisilleen tarjoamissa palveluissa. Palveluiden kysynnän kasvun ja monipuolistumisen myötä tietojärjestelmien välisen yhteentoimivuuden toteuttamisesta on kehittymässä entistäkin suurempi haaste taloudelliselle kasvulle.

Puhtaasti syntaktiset menetelmät järjestelmien välisen semanttisen yhteentoimivuuden toteuttamiseksi tekevät liian paljon virheitä. Semanttinen yhteentoimivuus edellyttää eri lähteistä peräisin olevan tiedon merkityksen lähes virheetöntä tunnistamista. Järjestelmien välisen semanttisen yhteentoimivuuden toteuttamisessa tarvitaan siksi sisällönkuvailua käsitteiden avulla.

Nykyisten tietojärjestelmissä käytettyjen käsitteistöjen ongelma on se, että vaikka eri käsitteistöt kuvailisivat monelta osin samoja tai osin päällekkäisiä kategorioita, tätä ei pystytä automaattisesti tunnistamaan. Semanttisen yhteentoimivuuden toteuttamiseksi nykyisin käytettyjen käsitteistöjen välille on siksi määriteltävä ihmisvoimin eksplisiittisiä semanttisia suhteita. Tästä aiheutuu merkittävä osa semanttisen yhteentoimivuuden toteuttamisen työläydestä ja suurista kustannuksista.

Tässä tutkielmassa esiteltiin ja tarkasteltiin semanttisen yhteentoimivuuden ratkaisemiseksi kuvauslogiikka+primitiivikäsitteistö-menetelmää. Tästä menetelmästä tarkasteltiin erityisesti OWL+SUMO-ilmentymää, jossa kuvauslogiikkana käytetään OWL-kuvauslogiikkaa ja primitiivikäsitteistönä SUMO-yläontologiaa.

Osoittautui, että OWL+SUMO-menetelmällä on mahdollista määritellä käsitteitä kuvailemaan erilaisia kohteita, esimerkiksi tilastointia tai yleisemmin tiedonhaku varten. Lisäksi osoittautui, että jos toimijat määrittelevät tarvitsemansa käsitteet tällä menetelmällä, niin eri toimijoiden määrittelemien käsitteiden semanttiset sisältymissuhteet voidaan löytää automaattisesti

Semanttisten suhteiden automaattinen tunnistaminen perustuu kuvauslogiikoissa formaalisti määriteltyyn tulkintafunktion, primitiivikäsitteisiin ja niiden yksikäsitteisiin tulkintoihin. Jos toimijat määrittelevät käsitteensä kuvauslogiikalla, käyt-

täen samaa primitiivikäsitteistöä tai primitiivikäsitteistöä, josta on määritelty merkitysvastaavuussuhteet toisten käyttämiin primitiivikäsitteistöihin, niin toimijoiden määrittelemien käsitteiden väliset semanttiset sisältymissuhteet voidaan löytää automaattisesti, standardipäätelyn toteuttavilla päättelykoneilla. Toimijat voivat siis määritellä primitiivikäsitteillä tarvitsemiaan käsitteitä vapaasti, toisistaan riippumattomasti, ja säilyttää edelleen semanttisen yhtenäisyyden.

Ihmisen muistissa kategoriaan kuulumiseen liittyy epävarmuutta ja riippumattomuutta yksittäisistä piirteistä. Klassiseen teoriaan perustuvalla kuvauslogiikalla tällaista käsitteisiin ihmisen muistissa liittyvää sumeutta ei ole mahdollista kuvailla. Niinpä kuvauslogiikalla voi monissa tapauksissa olla hyvin vaikeata määritellä käsitteitä kategoriaan kuulumiselle vaadittavina täsmällisinä välttämättöminä ja riittävinä ehtoina.

Käsitteiden määrittely kuvauslogiikka+primitiivikäsitteistö-menetelmällä vaatii kuvauslogiikan hallitsemisen lisäksi käytettävän primitiivikäsitteistön melko perusteellista tuntemista. Muita OWL+SUMO-menetelmän käytännön soveltamisessa vastaan tulevia haasteita voivat olla sen ilmaisuvoiman rajoittuneisuus ja OWL-kuvauslogiikalla esitetystä tietämyksestä mahdollisesti esiintyvien päättelyongelmien pahimman tapauksen suuri aikavaativuus.

## Lähteet

- AG06 Jürgen Angele and Michael Gesmann. Data integration using semantic technology: A use case. In *Proceedings of the 3rd European Semantic Web Conference (ESWC2006)*, Budva, Montenegro, Jun 2006.
- All84 James Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
- BC02 Ted Briscoe and John Carroll. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504, Gran Canaria, Las Palmas, 2002.
- BCM<sup>+</sup>03a Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- BCM<sup>+</sup>03b Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter 1. Cambridge University Press, 2003.
- BCM<sup>+</sup>03c Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter 2. Cambridge University Press, 2003.
- BGM96 Stefano Borgo, Nicola Gurrarino, and Claudio Masalo. A pointless theory of space based on strong connection and congruence. In *Proceedings of the fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pages 220–229, Cambridge, Massachusetts, USA, Nov 1996.
- Bor95 Alexander Borgida. Description logics in data management. *Knowledge and Data Engineering*, 7(5):671–682, 1995.

- BS01 Franz Baader and U. Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69:5–40, 2001.
- BTMS04 Michael Bada, Daniele Turi, Robin McEntire, and Robert Stevens. Using reasoning to guide annotation with gene ontology terms in goat. *SIGMOD Rec.*, 33(2):27–32, 2004.
- BWL<sup>+</sup>05 Rafael Batres, Matthew West, David Leal, David Price, and Yuji Naka. Some industrial experiences in the development and use of ontologies. In *The European Symposium on Computer Aided Process Engineering (ESCAPE)*, May 2005.
- CV95 Roberto Casati and Achille C. Varzi. *Holes and other superficialities*. MIT Press, Aug 1995.
- dBLPF05 Jos de Bruijn, Rubén Lara, Axel Polleres, and Dieter Fensel. OWL DL vs. OWL Flight: Conceptual modeling and reasoning for the semantic web, 2005.
- FBS02 Ian Horrocks Franz Baader and U. Sattler. Description logics for the semantic web. *DI - Kunstliche Intelligenz*, 16(4):57–59, April 2002.
- FGV01 Pasquale Foggia, Roberto Genna, and Mario Vento. Symbolic vs. connectionist learning: An experimental comparison in a structured domain. *IEEE Trans. Knowl. Data Eng.*, 13(2):176–195, 2001.
- GHT06 Tom Gardiner, Ian Horrocks, and Dmitry Tsarkov. Automated benchmarking of description logic reasoners. In *Proc. of the 2006 Description Logic Workshop (DL 2006)*, volume 189. CEUR (<http://ceur-ws.org/>), 2006.
- GHVD03 B. Groszof, Ian Horrocks, R. Volz, and S. Decker. Description logic programs: Combining logic programs with description logic. In *Proceedings of 12th International Conference on the World Wide Web (WWW-2003)*, 2003.

- GLP05 Michael Gallaher, Albert Link, and Jeffrey Petrusa. Measuring service-sector research and development. Technical Report 08236.002.004, RTI International, 2005.
- Gru93 Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.
- GS05 Robert L. Goldstone and Ji Yun Son. Similarity. In Keith J. Holyoak and Robert G. Morrison, editors, *The Cambridge Handbook of Thinking and Reasoning*, chapter 2. Cambridge University Press, 2005.
- HE04 Lewis Hart and Patrick Emery. A description logic for use as the odm core. In *Proceedings of the The 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC2004)*, 2004.
- Hor98 Ian Horrocks. Using an expressive description logic: FaCT or fiction? In Anthony G. Cohn, Lenhart Schubert, and Stuart C. Shapiro, editors, *KR'98: Principles of Knowledge Representation and Reasoning*, pages 636–645. Morgan Kaufmann, San Francisco, California, 1998.
- Hor02 Ian Horrocks. DAML+OIL: a description logic for the semantic web. *IEEE Data Engineering Bulletin*, 25(1):4–9, 2002.
- Hov05 Eduard H. Hovy. Methodologies for the reliable construction of ontological knowledge. In *ICCS*, pages 91–106, 2005.
- HPSvH03 Ian Horrocks, P. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.
- IEE90 IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries, 1990.
- ISO03 Industrial automation systems and integration - integration of life-cycle data for process plants including oil and gas production facilities - part 2: Data model, 2003.

- ISO05a Industrial automation systems and integration – integration of life-cycle data for process plants including oil and gas production facilities – part 4: Initial reference data, 2005.
- ISO05b Industrial automation systems and integration – integration of life-cycle data for process plants including oil and gas production facilities – part 7: Implementation methods for data exchange and integration, 2005.
- kan06 Hallituksen strategia-asiakirjan 2005 vaikuttavuusarviointi, 2006. Valitioseuvoston kanslian raportteja 4/2006.
- KS05 Gunnar O. Klein and Barry Smith. Concept systems and ontologies: Recommendations based on discussions between realist philosophers and iso/cen experts concerning the standards addressing "concepts" and related terms, 2005.
- Lei03 Jochen L. Leidner. Current issues in software engineering for natural language processing. In *SEALTS '03: Proceedings of the HLT-NAACL 2003 workshop on Software engineering and architecture of language technology systems*, pages 45–50, Morristown, NJ, USA, May 2003. Association for Computational Linguistics.
- LM01 O. Lassila and D. McGuinness. The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information Science*, 6(5), 2001.
- LS05 Robyn A. LeBoeuf and Eldar B. Shafir. Decision making. In Keith J. Holyoak and Robert G. Morrison, editors, *The Cambridge Handbook of Thinking and Reasoning*, chapter 11. Cambridge University Press, 2005.
- MHN04 R. Moller, V. Haarslev, and B. Neumann. Expressive description logics for agent-based information retrieval. In Y J. Cuenca, Demazeau, A. Garcia, and J. Treur, editors, *Knowledge Engineering and Agent Technology*, Volume 52, Frontiers in Artificial Intelligence and Applications. IOS Press, 2004.

- Min74 Marvin Minsky. A framework for representing knowledge. Technical Report 306, Artificial Intelligence Laboratory, MIT, 1974.
- MR05 Douglas L. Medin and Lance J. Rips. Concepts and categories: Memory, meaning, and metaphysics. In Keith J. Holyoak and Robert G. Morrison, editors, *The Cambridge Handbook of Thinking and Reasoning*, chapter 3. Cambridge University Press, 2005.
- Mur02a Gregory L. Murphy. *The Big Book of Concepts*, chapter 2. The MIT Press., 2002.
- Mur02b Gregory L. Murphy. *The Big Book of Concepts*, chapter 1. The MIT Press., 2002.
- Mur02c Gregory L. Murphy. *The Big Book of Concepts*, chapter 7. The MIT Press., 2002.
- Nii96 Ilkka Niiniluoto. *Informaatio, tieto ja yhteiskunta. Filosofinen käsiteanalyysi*. Valtion painatuskeskus ja Valtionhallinnon kehittämiskeskus, Helsinki, 1996.
- Nii97 Ilkka Niiniluoto. *Johdatus tieteenfilosofiaan*, chapter 3.4. Otava, Keuruu, 1997.
- NP01 Ian Niles and Adam Pease. Towards a standard upper ontology. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9, New York, NY, USA, 2001. ACM Press.
- NT03 Deborah Nichols and Allan Terry. User's guide to teknowledge ontologies, Dec 2003. [http://ontology.teknowledge.com/Ontology\\_User\\_Guide.doc](http://ontology.teknowledge.com/Ontology_User_Guide.doc) [12.1.2007].
- OAH+06 Daniel Oberle, Anupriya Ankolekar, Pascal Hitzler, Philipp Cimiano, Michael Sintek, Malte Kiesel, B. Mougouie, S. Vembu, S. Baumann, Massimo Romanelli, Paul Buitelaar, R. Engel, D. Sonntag, N. Reithinger, Berenike Loos, R. Porzel, H.-P. Zorn, V. Micelli, C Schmidt, Moritz



- Weiten, F. Burkhardt, and J. Zhou. DOLCE ergo SUMO: On foundational and domain models in swinto (smartweb integrated ontology). Technical report, AIFB, University of Karlsruhe, Jul 2006.
- Oja05a Matti Ojala. Palveluluokitus. projektin loppuraportti (luonnos), 2005.
- Oja05b Matti Ojala. Sosiaali- ja terveydenhuollon palveluluokitus versio 3.0, Mar 2005.
- OPB05 Alessandro Oltramari, Laurent Prévot, and Stefano Borgo. Theoretical and practical aspects of interfacing ontologies and lexical resources. In *Proceedings of SWAP 2005*, Trento, Italy, Dec 2005.
- OST00 The service economy, 2000. Final Report of the Business and Industry Policy Forum on Realising the Potential of the Service Economy.
- Pan05 Zhengxiang Pan. Benchmarking dl reasoners using realistic ontologies. In *Proceedings of the OWLED\*05 Workshop on OWL: Experiences and Directions*, Galway, Ireland, Nov 2005.
- PR04 Jinsoo Park and Sudha Ram. Information systems interoperability: What lies beneath? *ACM Trans. Inf. Syst.*, 22(4):595–632, 2004.
- Rec02 Alan Rector. Normalisation of ontology implementations: Towards modularity, re-use, and maintainability. In *Workshop on Ontologies for Multiagent Systems (OMAS) in conjunction with European Knowledge Acquisition Workshop*, Sigüenza, Spain, 2002.
- Rec03 Alan Rector. Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In *K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture*, pages 121–128, New York, NY, USA, 2003. ACM Press.
- RN95 Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Pacific Grove, CA, USA, 1995.
- RN03 Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.

- Rog06     Jeremy Rogers. Quality assurance of medical ontologies. *Methods of Information in Medicine*, 45, 2006. in press.
- RR05     Alan Rector and Jeremy Rogers. Ontological & practical issues in using a description logic to represent medical concepts: Experience from GALEN. Technical Report CSPP-35:1-35, "School of Computer Science, University of Manchester", 2005.
- Smi96     Barry Smith. Mereotopology: A theory of parts and boundaries. *Data & Knowledge Engineering*, 20(3):287–303, 1996.
- Smi06     Barry Smith. Against idiosyncrasy in ontology development. In *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS 2006)*, Baltimore, Maryland, USA, Nov 2006.
- Sow00     John F. Sowa. *Knowledge representation: logical, philosophical and computational foundations*, chapter 3.2. Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 2000.
- SPG<sup>+</sup>05     Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. Technical Report 2005-68, UMIACS, 2005.
- SR06     Julian Seidenberg and Alan Rector. Web ontology segmentation: analysis, classification and use. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 13–22, New York, NY, USA, 2006. ACM Press.
- Sun01     Ron Sun. Artificial intelligence: Connectionist and symbolic approaches. In N. J. Smelser and P. B. Baltes, editors, *International Encyclopedia of the Social and Behavioral Sciences*, pages 783–789. Pergamon/Elsevier, Oxford, 2001.
- Tho91     Chris. J. Thornton. Introduction to connectionist computing. *Computer-Aided Design*, 23(8):530–538, 1991.

- TL01 A. M. Tam and C. H. C. Leung. Structured natural-language descriptions for semantic content retrieval of visual materials. *J. Am. Soc. Inf. Sci. Technol.*, 52(11):930–937, 2001.
- Vie02a Stakes Viestintä. Sosiaali- ja terveydenhuollon asiakas- ja potilasasiakirjasanasto, 2002. [http://sty.stakes.fi/NR/rdonlyres/0C799961-C977-45DE-B0F6-7C92DA1D38B9/4016/http\\_\\_\\_wwwstakesfi\\_oske\\_terminologia\\_sanastot\\_aspo.pdf](http://sty.stakes.fi/NR/rdonlyres/0C799961-C977-45DE-B0F6-7C92DA1D38B9/4016/http___wwwstakesfi_oske_terminologia_sanastot_aspo.pdf) [12.1.2007].
- Vie02b Stakes Viestintä. Sosiaali- ja terveydenhuollon käsitteitä tietojärjestelmien suunnittelua varten, 2002. [http://sty.stakes.fi/NR/rdonlyres/0C799961-C977-45DE-B0F6-7C92DA1D38B9/4017/http\\_\\_\\_wwwstakesfi\\_oske\\_terminologia\\_sanastot\\_kasi.pdf](http://sty.stakes.fi/NR/rdonlyres/0C799961-C977-45DE-B0F6-7C92DA1D38B9/4017/http___wwwstakesfi_oske_terminologia_sanastot_kasi.pdf) [12.1.2007].
- VMS Professional herpetoculture for the pet trade. <http://www.cs.umbc.edu/kqml/kqmlspec.ps> [31.7.2006].
- W3Ca Naming and addressing: URIs, URLs, ... <http://www.w3.org/Addressing/> [23.8.2006].
- W3Cb OWL web ontology language guide. <http://www.w3.org/TR/owl-guide/> [29.8.2006].
- W3Cc OWL web ontology language overview. <http://www.w3.org/TR/owl-features/> [28.8.2006].
- W3Cd Resource description framework (RDF): Concepts and abstract syntax. <http://www.w3.org/TR/rdf-concepts/> [23.8.2006].
- W3Ce Web characterization terminology & definitions sheet. <http://www.w3.org/1999/05/WCA-terms/> [23.8.2006].
- W3Cf XML schema datatypes in RDF and OWL. <http://www.w3.org/TR/swbp-xsch-datatypes/> [29.10.2006].

- W3Cg XML schema part 2: Datatypes second edition. <http://www.w3.org/TR/xmlschema-2/> [29.10.2006].
- Way97 E. C. Way. Connectionism and conceptual structure. *American Behavioral Scientist*, 40:729–753, May 1997.
- Wes96 Matthew West. Developing high quality data models (version 2.0). Technical report, Shell International Limited, ISCL/4, Shell Centre, London, SE1 7NA, UK, 1996.
- Wes03 Matthew West. Common reference data – the foundation of e-business. In *Product Data Technology Europe 2003*, pages 159–166, 2003.
- Wes04 Matthew West. Some industrial experiences in the development and use of ontologies. In *International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004)*, Northamptonshire, UK, Oct 2004.
- Woo75 W. A. Woods. What’s in a link: Foundations for semantic networks. In D. G. Bobrow and A. Collins, editors, *Representation and Understanding*, pages 35–82. Academic Press, New York, 1975.
- Wro06 Chris Wroe. Is semantic web technology ready for healthcare. In *Proceedings of the 3rd European Semantic Web Conference (ESWC2006)*, Budva, Montenegro, Jun 2006.
- WSGA03 C. Wroe, R. Stevens, C. Goble, and M. Ashburner. A methodology to migrate the gene ontology to a description logic environment using DAML+OIL. In *2003 Pacific Symposium on Biocomputing Proceedings (PSB)*, USA, Hawaii, Jan 2003.

# Liite 1. Esimerkkikäsitteiden määrittelyt sarjallistettuna XML-esitysmuotoon

Alla on kappaleessa 7 määritellyt käsitteet sarjallistettuna XML-esitysmuotoon. Käsitteisiin viitataan termien sijaan yksilöivillä URI-tunnisteilla.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE rdf:RDF [
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
<!ENTITY sumo "http://reliant.teknowledge.com/DAML/SUMO.owl#">
<!ENTITY tietotyypit 'http://www.cs.helsinki.fi/u/pmlindgr/ontologies/tietotyypit.xsd#'>
]>
<rdf:RDF
  xmlns="http://www.stakes.fi/oske/luokitukset/palveluluokitus/palveluluokitus.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:sumo="http://reliant.teknowledge.com/DAML/SUMO.owl#"
  xml:base="http://www.stakes.fi/oske/luokitukset/palveluluokitus/palveluluokitus.owl">

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

  <owl:ObjectProperty rdf:ID="hasSkillOf">
    <owl:inverseOf rdf:resource="&sumo;hasSkill" />
  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="locationOf">
    <owl:inverseOf rdf:resource="&sumo;located" />
  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="timeOf">
    <owl:inverseOf rdf:resource="&sumo;time" />
  </owl:ObjectProperty>

  <owl:Class rdf:ID="Palvelu">
    <rdfs:label xml:lang="fi">Palvelu</rdfs:label>
    <rdfs:subClassOf rdf:resource="&sumo;SocialInteraction" />
  </owl:Class>

  <owl:Class rdf:ID="HoitoalanAmmattihenkilo">
    <rdfs:label xml:lang="fi">HoitoalanAmmattihenkilo</rdfs:label>
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="&sumo;Human" />
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasSkillOf" />
  <owl:someValuesFrom rdf:resource="&sumo;TherapeuticProcess" />

```

```

</owl:Restriction>
  </owl:intersectionOf>
  </owl:Class>
</owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Ammattihoito">
  <rdfs:label xml:lang="fi">Ammattihoito</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="&sumo;TherapeuticProcess" />
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;agent" />
  <owl:someValuesFrom rdf:resource="#HoitoalanAmmattihenkilo" />
</owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Tapaturmapaikka">
  <rdfs:label xml:lang="fi">Tapaturmapaikka</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;Region" />
<owl:Restriction>
  <owl:onProperty rdf:resource="#locationOf" />
  <owl:someValuesFrom rdf:resource="&sumo;Injuring" />
</owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="HoitopaikkaanSiirtyminen">
  <rdfs:label xml:lang="fi">HoitopaikkaanSiirtyminen</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;Translocation" />
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;destination" />
  <owl:someValuesFrom rdf:resource="#Hoitopaikka" />
</owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="HoitopaikkaanSiirtymisenAika">
  <rdfs:label xml:lang="fi">HoitopaikkaanSiirtymisenAika</rdfs:label>

```

```

    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="&sumo;TimeInterval" />
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Class rdf:ID="HoitopaikkaanSiirtymisenAikainen">
    <rdfs:label xml:lang="fi">HoitopaikkaanSiirtymisenAikainen</rdfs:label>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&sumo;time" />
        <owl:someValuesFrom rdf:resource="&sumo;HoitopaikkaanSiirtymisenAika" />
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Class rdf:ID="TapaturmapaikallaOleva">
    <rdfs:label xml:lang="fi">TapaturmapaikallaOleva</rdfs:label>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&sumo;located" />
        <owl:someValuesFrom rdf:resource="&sumo;Tapaturmapaikka" />
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Class rdf:ID="HoitoonSiirtymisenAikainenTaiTapaturmapaikallaOleva">
    <rdfs:label xml:lang="fi">HoitoonSiirtymisenAikainenTaiTapaturmapaikallaOleva</rdfs:label>
    <owl:equivalentClass>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="&sumo;HoitopaikkaanSiirtymisenAikainen" />
          <owl:Class rdf:about="&sumo;TapaturmapaikallaOleva" />
        </owl:unionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Class rdf:ID="Hoitopaikka">
    <rdfs:label xml:lang="fi">Hoitopaikka</rdfs:label>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&sumo;locationOf" />
        <owl:someValuesFrom rdf:resource="&sumo;TherapeuticProcess" />
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>

```

```

<owl:Class rdf:ID="Ensihoito">
  <rdfs:label xml:lang="fi">Ensihoito</rdfs:label>
  <rdfs:comment>
    STAKES-Palveluluokitus: "Alan koulutuksen saaneen ammattihenkilön
    antama hoito tapahtumapaikalla tai ennen hoitopaikkaan saapumista".
  </rdfs:comment>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Ammattihoito" />
        <owl:Class rdf:about="#HoitoonSiirtymisenAikainenTaiTataturmapaikallaOleva" />
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Ensihoitopalvelu">
  <rdfs:label xml:lang="fi">Ensihoitopalvelu</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Ensihoito" />
<owl:Class rdf:about="#Palvelu" />
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Sairaankuljetus">
  <rdfs:label xml:lang="fi">Sairaankuljetus</rdfs:label>
  <rdfs:comment>
    STAKES-palveluluokitus: ''terveydenhuollon tai ensihoitoalan
    koulutuksen saaneen ammattihenkilön saaneen ammattihenkilöstön
    suorittama potilaan siirtäminen mm. ambulanssilla tai helikopterilla
    hoitopaikkaan.''.
  </rdfs:comment>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;Transfer" />
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;agent" />
  <owl:someValuesFrom rdf:resource="#HoitoalanAmmattihenkilo" />
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;patient" />
  <owl:someValuesFrom rdf:resource="#Sairas" />
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;destination" />
  <owl:someValuesFrom rdf:resource="#Hoitopaikka" />
</owl:Restriction>
<owl:Class rdf:about="#HoitopaikkaanSiirtymisenAikainen" />

```



```

    </owl:intersectionOf>
  </owl:Class>
</owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Sairas">
  <rdfs:label xml:lang="fi">Sairas</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;Human" />
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;property" />
  <owl:someValuesFrom rdf:resource="&sumo;DiseaseOrSyndrome" />
</owl:Restriction>

</owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Leikkaustoiminta-Lastenkirurgia">
  <rdfs:label xml:lang="fi">Leikkaustoiminta-Lastenkirurgia</rdfs:label>
  <rdfs:comment>
    STAKES-Palveluluokitus: -
  </rdfs:comment>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;Surgery" />
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;patient" />
  <owl:someValuesFrom rdf:resource="#Lapsi" />
</owl:Restriction>

</owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Lapsi">
  <rdfs:label xml:lang="fi">Lapsi</rdfs:label>
  <rdfs:comment>
    MeSH: Child: age 6-12 yr

    Määritelmä muutettu muotoon 0-12 vuotta, joka tuottaa päättelyyn paremmin
    arkijärkeä vastaavia tuloksia.
  </rdfs:comment>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;Human" />
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;age"/>
  <owl:allValuesFrom rdf:resource="#Alle13Vuotta"/>
</owl:Restriction>

```

```

    </owl:intersectionOf>
  </owl:Class>
</owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="AlleKouluikäinenLapsi">
  <rdfs:label xml:lang="fi">AlleKouluikäinenLapsi</rdfs:label>
  <rdfs:comment>
    MeSH: Child, Preschool: age 2-5 yr

    Määritelmä muutettu muotoon 0-5 vuotta, joka tuottaa päättelyyn paremmin
    arkijärkeä vastaavia tuloksia.
  </rdfs:comment>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;Human" />
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<rdfs:Datatype rdf:about="&tietotyyppit;alle13"/>
<rdfs:Datatype rdf:about="&tietotyyppit;alle6"/>

<owl:Class rdf:ID="Alle13Vuotta">
  <rdfs:label xml:lang="fi">Alle13Vuotta</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;YearDuration" />
        <owl:Restriction>
          <owl:onProperty rdf:resource="&sumo;MagnitudeFn" />
          <owl:someValuesFrom rdf:resource="&tietotyyppit;alle13"/>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Alle6Vuotta">
  <rdfs:label xml:lang="fi">Alle6Vuotta</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;YearDuration" />
        <owl:Restriction>
          <owl:onProperty rdf:resource="&sumo;MagnitudeFn" />
          <owl:someValuesFrom rdf:resource="&tietotyyppit;alle6"/>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

```

        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>

  <owl:ObjectProperty rdf:ID="parentOf">
    <owl:inverseOf rdf:resource="&sumo;parent" />
  </owl:ObjectProperty>

  <owl:Class rdf:ID="AlleKouluikaisenVanhempi">
    <rdfs:label xml:lang="fi">AlleKouluikaisenVanhempi</rdfs:label>
    <rdfs:comment xml:lang="fi">
      Vanhempi jolla on kouluikainen lapsi
    </rdfs:comment>
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="&sumo;Human" />
          <owl:Restriction>
            <owl:onProperty rdf:resource="#parentOf" />
            <owl:someValuesFrom rdf:resource="#AlleKouluikainenLapsi" />
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Class rdf:ID="Terveysneuvonta">
    <rdfs:label xml:lang="fi">Terveysneuvonta</rdfs:label>
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="&sumo;EducationalProcess" />
        </owl:Restriction>
        <owl:onProperty rdf:resource="&sumo;represents" />
        <owl:someValuesFrom rdf:resource="&sumo;BiologicalAttribute" />
      </owl:Restriction>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

  <owl:Class rdf:ID="AlleKouluikaisenTerveysneuvonta">
    <rdfs:label xml:lang="fi">AlleKouluikaisenTerveysneuvonta</rdfs:label>
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Terveysneuvonta" />
        </owl:Restriction>
        <owl:onProperty rdf:resource="&sumo;patient" />
        <owl:someValuesFrom rdf:resource="#AlleKouluikainenLapsi" />
      </owl:Restriction>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

```

    </owl:equivalentClass>
  </owl:Class>

  <owl:Class rdf:ID="AlleKouluikaisenVanhemmanTerveysneuvonta">
    <rdfs:label xml:lang="fi">AlleKouluikaisenVanhemmanTerveysneuvonta</rdfs:label>
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Terveysneuvonta" />
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Restriction>
    <owl:onProperty rdf:resource="&sumo;patient" />
    <owl:someValuesFrom rdf:resource="#AlleKouluikaisenVanhempi" />
  </owl:Restriction>
  </owl:intersectionOf>
  </owl:Class>
  </owl:equivalentClass>
</owl:Class>

  <owl:Class rdf:ID="AlleKouluikaisenTaiSellaisenVanhemmanTerveysneuvonta">
    <rdfs:label xml:lang="fi">AlleKouluikaisenTaiSellaisenVanhemmanTerveysneuvonta</rdfs:label>
    <owl:equivalentClass>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#AlleKouluikaisenTerveysneuvonta" />
          <owl:Class rdf:about="#AlleKouluikaisenVanhemmanTerveysneuvonta" />
        </owl:unionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Class rdf:ID="AlleKouluikaisenRokotuspalvelu">
    <rdfs:label xml:lang="fi">AlleKouluikaisenRokotuspalvelu</rdfs:label>
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="&sumo;Injecting" />
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Restriction>
    <owl:onProperty rdf:resource="&sumo;patient" />
    <owl:someValuesFrom rdf:resource="#AlleKouluikainenLapsi" />
  </owl:Restriction>
  </owl:intersectionOf>
  </owl:Class>
  </owl:equivalentClass>
</owl:Class>

  <owl:Class rdf:ID="AlleKouluikaisenTutkimuspalvelu">
    <rdfs:label xml:lang="fi">AlleKouluikaisenTutkimuspalvelu</rdfs:label>
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="&sumo;Investigating" />
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Restriction>
    <owl:onProperty rdf:resource="&sumo;patient" />
    <owl:someValuesFrom rdf:resource="#AlleKouluikainenLapsi" />
  </owl:Restriction>
  </owl:intersectionOf>
  </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

```

</owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Lastenneuvolapalvelu">
  <rdfs:label xml:lang="fi">Lastenneuvolapalvelu</rdfs:label>
  <rdfs:comment>
    STAKES-Palveluluokitus: "alle kouluikäisille lapsille ja vanhemmille
    annettava terveysneuvonta, joka sisältää myös lapsen tutkimus- ja
    rokotuspalveluja."
  </rdfs:comment>
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#AlleKouluikäisenTaiSellaisenVanhemmanTerveysneuvonta" />
<owl:Class rdf:about="#AlleKouluikäisenRokotuspalvelu" />
<owl:Class rdf:about="#AlleKouluikäisenTutkimuspalvelu" />
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Leikki">
  <rdfs:label xml:lang="fi">Leikki</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="&sumo;RecreationOrExercise" />
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;agent" />
  <owl:someValuesFrom rdf:resource="#Lapsi" />
</owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="LeikinOhjaus">
  <rdfs:label xml:lang="fi">LeikinOhjaus</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="&sumo;Guiding" />
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;patient" />
  <owl:someValuesFrom rdf:resource="#Leikki" />
</owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

```

<owl:Class rdf:ID="Lastenhoito">
  <rdfs:label xml:lang="fi">Lastenhoito</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="&sumo;Maintaining" />
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;patient" />
  <owl:someValuesFrom rdf:resource="#Lapsi" />
</owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:ObjectProperty rdf:ID="startsBy">
  <owl:inverseOf rdf:resource="&sumo;starts" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="finishesBy">
  <owl:inverseOf rdf:resource="&sumo;finishes" />
</owl:ObjectProperty>

<owl:Class rdf:ID="MaanantaistaPerjantaihinViikolla">
  <rdfs:label xml:lang="fi">MaanantaistaPerjantaihinViikolla</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;TimeInterval" />
<owl:Restriction>
  <owl:onProperty rdf:resource="#startsBy" />
  <owl:someValuesFrom rdf:resource="&sumo;Monday" />
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#finishesBy" />
  <owl:someValuesFrom rdf:resource="&sumo;Friday" />
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;during" />
  <owl:allValuesFrom rdf:resource="&sumo;Week" />
</owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="OsanPaivasta">
  <rdfs:label xml:lang="fi">OsanPäivästä</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty rdf:resource="&sumo;temporalPart" />
          <owl:someValuesFrom rdf:resource="&sumo;Day" />
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

```

    </owl:Restriction>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&sumo;temporalPart" />
    <owl:allValuesFrom rdf:resource="&sumo;Day" />
  </owl:Restriction>
</owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="MaanantaistaPerjantaihinOsanPaivasta">
  <rdfs:label xml:lang="fi">MaanantaistaPerjantaihinOsanPäivästä</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#MaanantaistaPerjantaihinViikolla" />
<owl:Class rdf:about="#OsanPaivasta" />
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="LeikinOhjausTaiLastenhoito">
  <rdfs:label xml:lang="fi">LeikinOhjausTaiLastenhoito</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#LeikinOhjaus" />
<owl:Class rdf:about="#Lastenhoito" />
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="Leikkitoiminta">
  <rdfs:label xml:lang="fi">Leikkitoiminta</rdfs:label>
  <rdfs:comment>
    STAKES: "lapsille suunnattu leikin ja toiminnan ohjausta ja valvontaa
    sisältävä lasten päivähoito."
  </rdfs:comment>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#LeikinOhjausTaiLastenhoito" />
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;time" />
  <owl:someValuesFrom rdf:resource="#MaanantaistaPerjantaihinOsanPaivasta" />
</owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="LastenKotihoidonTuki">

```

```

<rdfs:label xml:lang="fi">LastenKotihoidonTuki</rdfs:label>
<rdfs:comment>
  STAKES: "kunnallisen päivähoidon vaihtoehtona maksettava sosiaaliavustus
  lapsen hoitamiseksi kotonaan. Sisältää myös mahdollisen lasten kotihoidon
  tuen kuntalisän."
</rdfs:comment>
<owl:equivalentClass>
  <owl:Class>
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:about="&sumo;Funding" />
      <owl:Restriction>
        <owl:onProperty rdf:resource="&sumo;destination" />
        <owl:someValuesFrom rdf:resource="#LapsenVanhempi" />
      </owl:Restriction>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&sumo;patient" />
        <owl:someValuesFrom rdf:resource="&sumo;LastenKotihoito" />
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
</owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="LapsenVanhempi">
  <rdfs:label xml:lang="fi">LapsenVanhempi</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="&sumo;Human" />
        <owl:Restriction>
          <owl:onProperty rdf:resource="#parentOf" />
          <owl:someValuesFrom rdf:resource="#Lapsi" />
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<owl:Class rdf:ID="LastenKotihoito">
  <rdfs:label xml:lang="fi">LastenKotihoito</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Lastenhoito" />
      </owl:intersectionOf>
    </owl:Class>
  <owl:Restriction>
    <owl:onProperty rdf:resource="&sumo;located" />
    <owl:someValuesFrom rdf:resource="&sumo;SingleFamilyResidence" />
  </owl:Restriction>
</owl:equivalentClass>
</owl:Class>

</rdf:RDF>

```



## Liite 2. Toimijan määrittelemät kyselykäsitteet sarjallistettuna XML-esitysmuotoon

Alla on kappaleessa 7 kuvitteellisen toimijan  $t_2$  määrittelemä kyselykäsite ja sen merkitykseen sisältyvät primitiivisemmät käsitteet sarjallistettuna XML-esitysmuotoon. Käsitteisiin viitataan termien sijaan yksilöivillä URI-tunnisteilla.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE rdf:RDF [
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY sumo "http://reliant.teknowledge.com/DAML/SUMO.owl#">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY datatypes "http://www.cs.helsinki.fi/u/pmlindgr/ontologies/datatypes.xsd#">
]>
<rdf:RDF
  xmlns="http://www.seco.tkk.fi/testi/queryConcepts.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:sumo="http://reliant.teknowledge.com/DAML/SUMO.owl#"
  xml:base="http://www.seco.tkk.fi/testi/queryConcepts.owl">

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

  <owl:Class rdf:ID="Child-targeted">
    <rdfs:label xml:lang="en">Child-targeted</rdfs:label>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&sumo;patient" />
        <owl:someValuesFrom rdf:resource="#Child" />
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Class rdf:ID="Child">
    <rdfs:label xml:lang="en">Child</rdfs:label>
    <rdfs:comment>
      alle 18-vuotias ihminen.
    </rdfs:comment>
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="&sumo;Human" />
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>

  <owl:Restriction>
    <owl:onProperty rdf:resource="&sumo;age"/>
    <owl:allValuesFrom rdf:resource="#Under18Years"/>
  </owl:Restriction>
</rdf:RDF>
```

```

    <owl:maxCardinality rdf:datatype="&xsd:int">1</owl:maxCardinality>
  </owl:Restriction>
    </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<rdfs:Datatype rdf:about="&datatypes;lessThan18"/>

<owl:Class rdf:ID="Under18Years">
  <rdfs:label xml:lang="fi">Under18Years</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="&sumo;YearDuration" />
<owl:Restriction>
  <owl:onProperty rdf:resource="&sumo;MagnitudeFn" />
  <owl:someValuesFrom rdf:resource="&datatypes;lessThan18"/>
</owl:Restriction>
    </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

</rdf:RDF>

```