

Käsitteellinen mallintaminen oppisisältöjen kuvaamisessa

Tampereen yliopisto
Tietojenkäsittelytieteen laitos
Pro gradu-tutkielma
Marketta Hölttä
Syyskuu 2004

Saatteeksi

Pro gradu -tutkielmani aihe oli alunperin matematiikan verkko-opetuksen kehittäminen Hämeen ammattikorkeakoulussa (HAMK). Työ alkoi syksyllä 2001 nykytilan selvityksellä, johon sisältyi matemaattisia aineita sisältävien opintojaksojen, opetuksen käytössä olevien verkkomateriaalien sekä matemaattisten aineiden opettajien verkkomateriaalien tuottamisvalmiuksien kartoitus. Sen pohjalta hahmoteltiin tarpeet suhteessa HAMK:n tieto- ja viestintätekniiikan opetuskäytön strategian tavoitteisiin. Tärkeimmäksi kehityskohteeksi nousi materiaalien löydettävyyden ja uudelleenkäytön mahdollistavan systeemin luominen. Tämä puolestaan johti oppisisältö-käsitteen analysointiin ja samanaikaisesti valmistuvassa olevan oppimisasihion metatietorakennestandardin (LOM) tutkimiseen. Tuolloin myös havaittiin, että samalla sovelluksella voidaan hallita minkä tahansa aihealueen oppimateriaaleja. Ainoastaan matemaattisten aineiden oppimateriaalien tuotanto vaatii erikoismerkkien vuoksi omat työkalunsa.

Ohjelmiston vaatimusten keruuvaiheessa törmättiin integrointiongelmiin. Verkkomateriaalien yksittäisten osien, sisältökomponenttien, käyttöä ja löytymistä hallinnoiva sovellus kaipasi ajantasaisia tietoja opetushenkilöstöstä, opintojaksoista ja niiden toteutuksista. Suunnitteilla olevan sovelluksen tuottamia tietoja pitäisi lisäksi voida tallentaa suoraan palkkahallinnon ja koulutuksen järjestelmiin. Suurimmaksi esteeksi järjestelmien yhteistoiminnallisuuden rakentamiselle havaittiin jo olemassa olevien järjestelmien käsitteellinen ja tekninen heterogeenisuus sekä niiden suljetut rajapinnat. Integrointi tulisi siis vaatimaan perusteellisempaa kohdealueen analyysia. Mielenkiintoinen ongelma vei mennessään. HAMK:ssa päätettiin tehdä perusteellinen tietojärjestelmien analyysi sekä siihen pohjautuen laatia mahdollisimman pitkälle tulevaisuuteen tietohallinnon toimintaa ohjaava tietohallintostrategia.

Nyt käsillä olevan tutkielman sisältöön on vaikuttanut järjestelmäintegraation ja sähköiseen asiointiin liittyvien taustateorioiden tutkiminen. Alkuperäisestä aiheesta ovat jäljellä osa nyt esitetystä käsitteellisen mallintamisen viitekehyksestä sekä siihen liittyen sisältökomponentti- ja oppimisasihiokäsitteiden tarkastelu.

Tiivistelmä

Osapuolten yhteinen kohteiden ja ilmiöiden käsitteellistäminen on edellytys ymmärrettävälle kommunikoinnille. Käsitteellinen mallintaminen on prosessi, joka tuottaa yhteisesti käytettyjen käsitteiden määritelmät. Sen avulla voidaan tuottaa ihmisen ymmärtämässä muodossa perustietämys ontologioiden, erityyppisten metatietojen sekä yhteisten käsitteiden esittämiseen. Tämän perustietämyksen pohjalta yhteinen käsitteellistäminen voidaan esittää jollakin keinotekoisella kielellä tietokoneohjelmien ymmärtämässä muodossa. Tällainen kieli näyttäisi tämän tutkielman perusteella olevan XML. Käsiteavaruus on ääretön. Siksi samasta kohteesta eri tarkoituksiin tehdyt käsitteelliset mallit poikkeavat toisistaan. Tutkin tässä tutkielmassa käsitteellisen mallintamisen prosessia, mallintamisen intensionaalista lähestymistapaa sekä käsitteellisen mallin, ontologian, metatietojen ja nk. sanastojen sisältöä ja keskinäistä suhdetta. Johtopäätöksenä voidaan todeta, että käsitteellinen mallintaminen kattaa sekä ontologian rakentamisen, metatietojen määrittämisen sekä sanastojen luontityön. Semanttiseen Webiin ja yleisesti ottaen tietosysteemien semantiikkaan liittyvän terminologian käyttö on hämmennystä aiheuttavaa ja vaatisi huomattavasti nykyistä tarkemman määrittelyn käyttöyhteydessään.

Avainsanat: käsitteellinen mallintaminen, käsitteellinen malli, ontologia, käsiterakenne, intensionaalinen sisältämssuhde, metadata, sanasto, Concept-D, LOM, XML.

Kuvaluettelo

- Kuva 1. Kohdealueen käsitteellistäminen ja tietosysteemi [29]
- Kuva 2. Semanttisten sovellusten kolmijako Dacontan mukaan [20]
- Kuva 3. Käsite rakenteet, joukkojen kuvaukset ja joukkojen ilmentymät [24]
- Kuva 4. Ontologiatyyppejä [13]
- Kuva 5. Sanaston, käsitteellistämisen, ontologisen sitoutumisen ja ontologian väliset suhteet [13]
- Kuva 6. Sanaston (tesaurus) ja ontologian suhde dacontan mukaan [20]
- Kuva 7. Käsitteellisen mallintamisen prosessi [24]
- Kuva 8. Käsitteellinen mallintaminen: teorian kehittäminen
- Kuva 9. Mallintamisen vaiheet
- Kuva 10. Ontologian ja tietosysteemien suhde
- Kuva 11. Aggregoimalla muodostettu johdettu käsite
- Kuva 12. Johdetun käsitteen määrittely aggregoimalla useasta käsitteestä
- Kuva 13. Kulkuneuvo-käsitteen evaluointi
- Kuva 14. Kulkuneuvo-käsite
- Kuva 15. Looginen tavoitearkkitehtuuri [38]
- Kuva 16. henkilöstö.xml
- Kuva 17. henkilöstö.xsd
- Kuva 18. henkilöstö.xml ja henkilöstö.xsc dokumenttien malli
- Kuva 19. Kurssi-käsite
- Kuva 20. Käsite rakenteesta ontologiaan.
- Kuva 21. Tavoitearkkitehtuuri, XML:n rooli [39].
- Kuva 22. Sanastojen ja sanastotyön tasot [40].
- Kuva 23. Julkishallinnon ohjeistuksen mukainen sanastojen kuvausmalli [40].
- Kuva 24. Perusydyntökomponentti [40].
- Kuva 25. Sanaston luontiprosessi [40].
- Kuva 26. HAMK:n tarkoituksiin määritelty oppimateriaalin sisältökomponentti-käsite.
- Kuva 27. LOM-standardin perusluokitus.
- Kuva 28. Sisältökomponentin ominaispiirteet/LOM:n perusluokitus.
- Kuva 29. General-kategoriassa kuvattavat ominaisuudet (LOM oppisisältö).
- Kuva 30. Sisältökomponentin ja LOMv1.0:n General-luokan synteesi.
- Kuva 31. Lifecycle kategorian käsitteellinen tietorakenne.
- Kuva 32. LOM:n Meta-Metadatan luokka.

Kuva 33. Meta-Metadatan luokan rakenne synteesin jälkeen.

Kuva 34. Technical-luokkaan kuuluvat ominaisuudet.

Kuva 35. Technical luokka synteesin jälkeen.

Kuva 36. Educational-kategorian ominaisuudet.

Kuva 37. Educational –luokka synteesin jälkeen.

Kuva 38. Rights-kategoriassa kuvattavat ominaisuudet.

Kuva 39. Relation luokan ominaisuudet.

Kuva 40. Annotation luokan ominaisuudet.

Kuva 41. Classification luokkaan luokitellut ominaisuudet.

Kuva 42. Classification luokan kansallisessa versiossa kuvattavat ominaisuudet.

SISÄLLYS

1. JOHDANTO	1
2. TUTKIMUKSEN TARKENTAMINEN	4
2.1 TUTKIMUKSEN TAVOITE JA ETENEMISEN LOGIIKKA.....	4
2.2 ESIMERKKIEN TAUSTAT	5
3. KÄSITTEELLISTÄMINEN JA KÄSITE	7
3.1 KÄSITE, KATEGORISOINTI JA KÄSITTEELLISET FUNKTIOT	7
3.2 KÄSITE TIETOJENKÄSITTELYSSÄ	10
3.3 YHTEENVETO	13
4. KÄSITTEELLINEN MALLINTAMINEN	15
4.1 ONTOLOGIOISTA, SANASTOISTA JA METATIEDOISTA.....	20
4.2 MALLINTAMISPROSESSI.....	31
4.3 YHTEENVETO	35
5. CONCEPT-D	38
5.1 PERUSKÄSITE JA SEN KUVAAMINEN	38
5.2 JOHDETTU KÄSITE JA SEN KUVAAMINEN.....	39
5.3 EHDOT JA RAJOITUKSET	42
5.4 TUNNISTEET	43
5.6 YHTEENVETO	44
6. XML	45
6.1 XML-DOKUMENTTI.....	46
6.2 XML-DOKUMENTIN SUUNNITTELU.....	49
6.3 KÄSITERAKENNE JA XML	51
7 JULKISHALLINNON SANASTOTYÖ	54
8.OPPISISÄLTÖ-KÄSITE JA SEN KUVAAMINEN	60
8.1. SISÄLTÖKOMPONENTTI.....	60
8.2 OPPISISÄLLÖN JA SISÄLTÖKOMPONENTIN VERTAILU	62
8.3 YHTEENVETO	71
9 JOHTOPÄÄTÖKSET	72
10 JATKOTUTKIMUKSEN AIHEITA	73
10.1 CONCEPT-D:N KÄYTTÖ XML-DOKUMENTTIEN JA SANASTOJEN MALLINNUKSESSA.....	73
10.2 JULKISHALLINNON SANASTOTYÖN TUTKIMINEN	73
10.3 LOM JA KÄYTÄNNÖN SOVELLUKSET	73
LÄHTEET	75
LIITE 1	77
LIITE 2A	79
LIITE 2B	80
LIITE 2C	81
LIITE 2D	82

1. Johdanto

Todellista hyötyä sekä asiakkaalle että organisaatiolle tuottavien verkkopalvelujen kehittäminen ei ole ongelmatonta. Ikääntyneitä perusjärjestelmiä ei pystytä sellaisenaan liittämään osaksi sähköisen asioinnin kokonaisuutta. Integroinnin ja siten tiedonkulun esteeksi saattavat muodostua järjestelmien suljetut, dokumentoimattomat rajapinnat. Järjestelmien yhteistoiminnallisuuden rakentamista hankaloittavat lisäksi niiden käsitteiden semanttinen ja toteutuksen tekninen heterogeenisuus. XML-merkintäkieleen [41] perustuvat tekniikat helpottavat integrointia, mutta eivät yksinään ratkaise ongelmia. XML-dokumentteja käsittelevillä järjestelmillä pitäisi lisäksi olla yhteinen tulkinta dokumentissa käytettyjen elementtien ja attribuuttien merkityksestä.

Semanttinen Web-visio on tuonut ontologia-termin yleiseen tietoisuuteen. Ontologia ei ole tietojenkäsittelyssä uusi asia, filosofiassa se on tunnettu Aristoteleen ajoista alkaen. 1970-luvun alussa tekoälyn, tietämyksen esittämisen sekä luonnollisen kielen prosessoinnin tutkijat havaitsivat älykkäitä systeemejä mahdollistavien prosessien suunnittelun ja toteutuksen vaativan pohjaksi kohdealueen ontologista analyysia. Nykyään ontologia-termi aiheuttaa liikettä mm. yritystoimintaan, e-kauppaan, XML:n metadatastandardien määrittämiseen sekä tiedon hallintaan ja löytymiseen kohdistuvaan tutkimukseen [12]. Lisäksi tiedon jakamisen ja vaihdon alueella ontologiasta on tullut oma sovellusalueensa. Duineveldin [6] mukaan syy ontologioiden kasvavaan suosioon löytyy tutkijoiden lupauksesta, jonka mukaan jostakin kohdealueesta voidaan luoda ihmisten ja koneiden kommunikoinnissaan käyttämä jaettu, yhteinen ymmärrys.

Tietokoneet saadaan haluttaessa tulkitsemaan asiat samalla tavalla. Käsitteiden joukko on kuitenkin ääretön. Käyttäjien kannalta on tärkeää löytää yhteinen sopimus käytetyistä käsitteistä ennenkuin niitä aletaan koodata jollakin formaalilla tavalla koneiden käsiteltäväksi. Yhteinen sopimus edellyttää kohdealueen käsitteiden tunnistamista, analysointia ja käsiterakenteiden tarkkaa kuvaamista.

Ontologian lisäksi Internetin kaaosta ja heterogeenisten järjestelmien aiheuttamia ongelmia pohtivassa ja niihin ratkaisua etsivässä kirjallisuudessa vilahtelevat usein termit käsitteellistäminen, käsitelmä, semanttinen informaatio, semantiikka, tietomalli, sanasto, olio ja luokka sekä monenlaiset meta-alkuiset termit. Uusien, koneen ymmärrettäviksi tar-

koitettujen teknologioiden omaksumista ja käyttöönottoa hankaloittaa suuresti niiden käsitteiden käytön epämääräisyys, joilla nk. älykkään webin tekniikoita perustellaan.

Useimmiten ontologia määritellään Gruberia [11] lainaten käsitteellistämisen spesifikaatioksi. Siitä puhutaan myös sisältöteorian, tekniikkana, sanastona tai käsitehierarkiana. Tutkin tässä työssäni käsitteellistä mallintamista ja siihen kiinteästi liittyvää kognitiotieteiden näkemystä käsitteiden muodostusteoriasta. Tältä pohjalta tarkastelen käytyä keskustelua ontologioista, semanttisesta informaatiosta, metatiedoista ja sanastoista. Tutkielman empiirisessä osiossa tutkin oppisisältöjen kansainvälistä metatietostandardia ja sen soveltuvuutta Hämeen ammattikorkeakoulun tarpeisiin.

Yhtenä keskeisenä elementtinä mm. valtion tietotekniikan rajapintasuosituksissa [38] on XML-standardi. XML:ään pohjautuvat tekniikat näyttäisivät olevan koko tulevaisuuden tietojenkäsittelyn ydintä. XML on metakieli rakenteisen tiedon kuvaamiseen. Rakenteistettu tieto mahdollistaa tietokokonaisuuksien osien kuvaamisen ja ohjelmallisen käsittelyn. Tietokokonaisuudet ovat useimmiten reaali maailman käsitteiden ilmentymiä. Concept-D on graafinen, 1980-luvulla kehitetty käsiterakenteiden kuvauskieli. Tämän työn punaisena lankana on kohdealueen luonnollisten käsitteiden mallintaminen osana tietosysteemien suunnittelutyötä. Mielenkiintoisena tehtävänä on tutkia perusfilosofialtaan hyvin samantyyppisten tiedon kuvauskielten, Concept-D:n ja XML:n toisiaan täydentävää roolia.

Luvussa 2 tarkennan tutkielman tarkoitusta ja sisältöä sekä esittelen lyhyesti esimerkkien taustat. Luvussa 3 tarkastelen käsitteiden muodostuksen teoriaa sekä käsitteisiin liittyviä määritelmiä. Luvussa 4 tutkin käsitteellistä mallintamista sekä ontologioista, metatiedoista ja sanastoista käytyä keskustelua sekä kuvaan mallintamisprosessin vaiheet. Luvut 6, 7 ja 8 käsittelevät käsitteiden kuvaamista siten, että sekä ihmiset että koneet voisivat muodostaa niistä yhteneväisen tulkinnan. Luvussa 6 tarkastelen mallintamisprosessin tuloksena syntyneiden käsiterakenteiden kuvaamista Concept-D-kielen kielellä. Luvussa 7 esittelen XML-ytimen perusteet sekä tarkastelen XML:n käytettävyyttä Concept-D:llä mallinnettujen käsitteiden kuvaamisessa. Luvussa 7 luon katsauksen julkishallinnon sanastotyön nimeämissuositusluonnokseen ts. tarkastelen elävän elämän esimerkkiä käsitteiden kuvaamisesta. Luvussa 8 tutkin samasta kohteesta, oppimateriaalin osasta, hieman eri tarkoituksiin muodostettua määritelmää ja vertaan niitä keskenään. Luvussa 9 kuvaan tutkielmani johtopäätökset ja luvussa 10 esittelen joitakin mielenkiintoisia jatkotutkimusaiheita.

2. Tutkimuksen tarkentaminen

2.1 Tutkimuksen tavoite ja etenemisen logiikka

Mitä oikeastaan tarkoitetaan metatiedolla tai ontologialla? Mikä on käsite ja miten se muodostuu? Onko semanttisella informaatiolla, käsitteellisellä mallilla tai sanastolla jotain tekemistä keskenään? Miten käsitteet voidaan määritellä ja esittää niin, että sekä koneet että ihmiset pystyisivät muodostamaan niistä yhteisen tulkinnan? Tämän työn tavoitteena on tutkia käsitteellisen mallintamisen ongelmakenttää ja metodeita sekä mallintamisen merkitystä rakennettaessa nykypäivän ja tulevaisuuden verkkopalveluita.

Ontologian useimmin käytetty määritelmä kuvaa sen käsitteellistämisen spesifikaatioksi. Kuten kaikki termit, myös käsitteellistäminen voidaan tulkita monella tavalla. Aloitan varsinaisen työni selvittämällä kognitiotieteiden tutkimustuloksia inhimillisestä käsitteenmuodostuksesta.

Tulevaisuuden verkkopalveluiden rakentamista pohtivassa kirjallisuudessa käytetty terminologia on aikalailla hämmentävää. Esimerkiksi ontologia määritellään myös sanastoksi, joka spesifioi kohdealueen käsitteet. Sanastoilla voidaan tarkoittaa suppeimmassa merkityksessään termien luetteloita ja laajimmassa merkityksessään käsitteiden systeemiä tai kokonaista käsitejärjestelmää. Metatietojen peittoalue on yhtä laaja. Metatiedoilla voidaan tarkoittaa esimerkiksi formaalissa muodossa esitettyjä ontologiakirjastoja tai vaikkapa palvelu- ja/tai prosessikuvauksia. Useimmiten se kuitenkin vielä määritellään tiedoksi tiedosta. Ennen varsinaisen mallintamisprosessin tutkimista pyrin luomaan katsauksen käsitteelliseen mallintamiseen olennaisesti sisältyvien termien käytettyihin merkityksiin.

Käsitteellinen mallintaminen voidaan karkeasti katsoa jakautuvan kahteen erilliseen koulukuntaan tutkittavan kohteen perusteella. Ekstensionaalisisessa lähestymistavassa tutkitaan käsitteiden ilmentymäjoukkoja kun taas intensionaalisisessa lähestymistavassa tutkitaan käsitteiden sisäistä rakennetta ja rakenteeseen sisältyvien määrittelevien käsitteiden välisiä suhteita. Tarkastelen tässä työssäni intensionaalisen lähestymistavan mukaista metodia ja esittelen siihen perustuvan graafisen käsiterakenteiden kuvauskielen, Concept-D:n. Tämän pohjalta tarkastelen Concept-D kielellä kuvattujen käsiterakenteiden koodaamista suoraan XML-dokumenteiksi.

Kohdealueen käsitteiden selvittäminen ja määrittäminen on perusedellytys käyttäjille hyötyä tuottavien verkkopalveluiden rakentamiselle, ohjelmistojen skaalautuvuudelle ja uudelleenkäytölle. Tietojenkäsittelyn kannalta mallintamisen tarkoituksena on kuvata kohdealue tavalla, jota sitä käsittelevät monenlaiset ohjelmistot voivat ymmärtää. Tällöin koneen tulkittaviksi tarkoitetut käsitteet pitää pystyä ilmaisemaan yksikäsitteisesti ensin ihmisten yhteisesti ymmärtämässä muodossa ja pyrkiä esittämään ne sitten koneen ymmärtämässä muodossa. Concept-D –kielen jälkeen esittelen lyhyesti XML-ytimen, johon perustuvat tekniikat näyttäisivät taipuvan siihen, että mallintamisen tulos voidaan muuntaa koneen ymmärtämään muotoon.

Mallintamisprosessin tulos on riippuvainen mallintamistehtävän tarkoituksesta, ts. samasta kohteesta eri tarkoituksiin tehdyt mallit poikkeavat toisistaan. Tutkielman empiirisessä osassa vertaan toisiinsa kahta samasta kohteesta, digitaalisesta oppimateriaalielementistä, miltei samoihin tarkoituksiin tehtyä käsittemallia toisiinsa. Vertailun kohteina ovat Hämeen ammattikorkeakoulussa määritelty digitaalisen oppimateriaalin osa, sisältökomponentti, sekä IEEE:n LTC:n standardi oppisisällön metatiedoista (Learning Object Metadata, LOM).

2.2 Esimerkkien taustat

HAMK – oppimateriaalin sisältökomponentti

Hämeen ammattikorkeakoulu (HAMK) on Kanta-Hämeessä ja Pirkanmaalla toimiva monialainen ammattikorkeakoulu, jolla on 11 opetuspistettä setsemällä eri paikkakunnalla. HAMK:ssa on viisi koulutusalaa, jotka sisältävät yhteensä 22 koulutusohjelmaa. Lisääntyvä ristiinopiskelu eri paikkakunnilla toteutettavien koulutusohjelmien välillä sekä yhteiset ammattikorkeakoulu- ja koulutusalaakohtaiset opinnot lisäävät digitaalisten oppimateriaalien ja verkko-opiskelun tarvetta. Oppimateriaalien uudelleenkäyttö, materiaalien löydettävyys ja tekijänoikeuksien turvaaminen edellyttävät jonkin systemaattisen oppimateriaalikomponenttien tallennus-, haku- ja käyttöjärjestelmän olemassaoloa. Tätä varten HAMK:ssa kuvattiin sisältökomponentti-käsite. Tarkoituksena oli kuvata oppimateriaalin osa niin tarkasti, että se kuvauksen perusteella voitaisiin valita yhdeksi rakennuspalikaksi rakennettaessa opintojaksoon liittyvää digitaalista oppimateriaalia. Tarve saman oppimateriaalin uudelleen- ja samanaikaiseen käyttöön eri opintojaksototeutuksissa syntyi havainnosta, jonka

mukaan esimerkiksi tilastomatematiikkaan sisältyviä, samaa aihetta käsitteleviä osioita opetettiin samanaikaisesti 13 opintojaksolla ammattikorkeakoulun eri koulutusohjelmissa.

IEEE/LTSC – Learning Object Metadata (LOM)

IEEE Learning Technology Standards Committee (LTSC) koostuu useista työ- ja tutkimusryhmistä. Niiden tehtävänä on kehittää teknisiä standardeja, suositeltavia käytäntöjä ja antaa oppimisympäristöjen kehitystä, ylläpitoa, hallintaa ja vuorovaikutteisuutta helpottavia ohjeistuksia. LTSC kokoaa suosituksia muilta standardointilaitoksilta ja määrittelee niistä ehdotukset ISO:n standardointikomitealle (IEC JTC1/SC36, Information Technology for Learning, Education, and Training). IEEE:n tutkimusryhmissä käsitellään mm. oppimisympäristöjen arkkitehtuureja ja sanastoja, opiskelijoihin, sisältöihin, dynaamisiin kurssirakenteisiin, metadataan ja systeemin hallintaan liittyviä tietomalleja, tietojenvaihdossa käytettäviä protokollia ja alustastandardeja. Learning Objects Metadata (LOM), IEEE 1474.12.1-2002 on standardi oppisisällön metatiedon kuvaukseen. Standardin tarkoitus on helpottaa oppijoiden, opettajien ja tietokoneohjelmien oppisisältöön kohdistuvaa etsimistä, arviointia, hankkimista ja käyttöä.

3. Käsitteellistäminen ja käsite

Ontologian yleisin määritelmä määrittelee sen käsitteellistämisen spesifikaatioksi. Kognitiivisen psykologian mukaan havaitsemiskyvyn eräs tärkeimmistä ominaisuuksista on se, että yksilöllisesti toisistaan eroavat kohteet, oliot ja tapahtumat kohdataan ajatuksissa ja kielessä käsitteellisten kategorioitten jäseninä. Kun biologian perusteista selvilläoleva suomenkielinen henkilö havaitsee esimerkiksi pentujaan imettävän koiran, hän tulkitsee eläimen nisäkkääksi ts. nisäkä-kategorian edustajaksi. Samat biologiset tiedot omaava englantinkielinen henkilö sijoittaa saman kohteen mammal-kategoriaan.

Frisby et al.:n mukaan käsitteet tai käsitteelliset kategoriat ovat muistiin varastoituja kohteiden, olioiden tai tapahtumien mentaalisia esityksiä. Täten kyky sijoittaa havaitsemamme kohteet käsitteellisiin kategorioihin on tajuamisen perusominaisuus [9]. Ellei meillä tätä ominaisuutta ole, emme pysty ajattelemaan - kommunikoinnista puhumattakaan. Myös muisti toimii tehottomasti ilman kykyä organisoida hankkimaamme tietoa tarkoitukseenmukaisiin kategorioihin.

Tässä luvussa tarkastelen ensin kognitiivisen psykologian näkemystä käsitteellistämisestä ja lopuksi tietojenkäsittelyn näkemyksiä käsitteisiin liittyvistä määritelmistä.

3.1 Käsite, kategorisointi ja käsitteelliset funktiot

Kohtaamalla esineet, oliot ja tapahtumat käsitteellisten kategorioitten jäseninä jaamme maailman erillisiin viipaleisiin. Tällä tavalla ikään kuin määräämme maailmamme rakenteen. Mutta mikä selittää juuri sellaisen rakenteen, jonka luomme?

Määritelmiä

Havaitun maailman rakenteeksi sanotaan periaatetta, jonka mukaan muodostamamme käsitteelliset kategoriat määräytyvät osittain maailmamme rakenteellisen luonteen ja osittain havaintomekanismiemme ominaisuuksien perusteella [9].

Kategorioiden ansiosta havaittu maailman rakenne voidaan esittää tavalla, joka tuottaa maksimaalisen hyödyn minimaalisilla ponnistuksilla. Tätä periaatetta kutsutaan *kognitiiviseksi ekonomiaksi* [9].

Ympäröivä kulttuuri puolestaan määrää sen, mitkä kategoriat täyttävät kognitiivisen ekonomian vaatimusperiaatteen. Muodostettujen kategorioiden pitää olla sellaisia, että yh-

teisön jäsenet pystyvät jakamaan ne keskenään. Ilmiötä sanotaan *pakollisen jaettavuuden periaatteeksi* [9].

Käsite on mentaalinen esitys tai idea, joka sisältää kategorian tärkeät ominaisuudet. Käsite viittaa kategoriaan, joka on määrätyn väitteen tai määrätty väitteet toteuttava ositus havaitusta maailmastamme. Jos kategoria sisältää useamman kuin yhden jäsenen, luokittelu koskee monenlaisia olioita, jotka ovat jollakin tapaa samanlaisia [8]. Käsitteellisten kategorioiden [9] *suhteet* ovat mentaalisesti esitettyjä hierarkioita, joissa yleisin ja ylimpänä oleva kategoria sisältää alemmilla tasoilla olevat, erikoisominaisuuksia sisältävät alikategoriat. Esimerkiksi väite, että nisäkkäällä on nisät, jaottelee elolliset oliot ainakin kahteen kategoriaan, joista nisäkäs-kategoria toteuttaa em. väitteen.

Wrobelin [46] mukaan kognitiivisen psykologian tutkimukset ovat osoittaneet, että havaitessaan kohteet ja ilmiöt ihmiset yhdistävät ne niiden ominaisuuksien perusteella käsitteellisiin kategorioihin. Ominaisuudet voivat olla monentyppisiä sisältäen havaintoihin perustuvat, toiminnalliset ja muodolliset ominaisuudet [9].

Frisbyn mukaan [9] välttämättömät ominaisuudet ovat niitä, jotka kohteella pitää olla, jotta se voidaan sisällyttää määrättyyn kategoriaan. Välttämättömät ja riittävät ominaisuudet määrittelevät *kategorian*. Käsitteelliset kategoriat, jotka voidaan määritellä käyttämällä välttämättömiä ja riittäviä ominaisuuksia kuvaavia termejä, ovat toisensa poissulkevia ja hyvinmääriteltyjä. Kategoriat, joilla ei ole välttämättömiä eikä riittäviä ominaisuuksia, ovat huonosti määriteltyjä ja sekavia, koska ne jakavat samoja ominaisuuksia yhdessä muiden kategorioiden kanssa. *Piirteet* ovat käsitteiden ominaisuuksia, jotka helpottavat vertailua ja selittävät samanlaisuutta. *Suhteet* osoittavat, miten käsitteen osat ja piirteet on yhdistetty rakennetta luotaessa. Yhteydet voivat olla avaruudellisia, ajallisia, syysuhteita, toiminnallisia tai teorioista johdettuja.

Käsitteen *intensio* sisältää sekä kategorian jäsenyyden määrittelevät kriteerit että ominaisuudet, jotka määrittävät käsitteen suhteet toisiin käsitteisiin. Käsitteen *ekstensio* on joukko olioita, joihin käsite viittaa. Käsitteen *ydin* koostuu sen oleellisista ominaisuuksista (attribuuteista) ja määritelmistä. *Idenfiointiproseduuri* on metodi käsitteen ilmentymien havaisemiseen ja tunnistamiseen.

Käsitteelliset funktiot

Käsitteellisten funktioiden [8] perusfunktio on luokittelu, jonka olemassaolo on edellytys muille käsitteellisille funktioille.

Luokittelu. Luokittelufunktio sisältää määrittelyn siitä, miksi jokin ilmentymä kuuluu määrättyyn kategoriaan ja miksi jokin erityinen käsite sisältyy toiseen käsitteeseen.

Ymmärtäminen ja selitys. Luokittelu mahdollistaa ihmisten ja älykkäiden systeemien jäsentävän kokemuksensa tarkoituksenmukaisesti lohkoihin ja muodostavan tulkinnan niistä. Tämän ymmärryksen päänäkökohta on tuoda vanha tietämys nykytilanteeseen.

Ennustaminen. Käsitteet ja käsiterakenteet mahdollistavat muistivarastot, joihin pohjautuen voidaan ennustaa tulevaisuutta.

Päätely. Käsitteet ja käsiterakenteet mahdollistavat päättelyn. Tällöin jokaista tosiasiaa ja mahdollisuutta ei tarvitse varastoida muistiin, jos ne voidaan johtaa jo varastoiduista tiedoista.

Kommunikointi. Kun ihmiset jakavat tiedon ja indeksoivat sen samaan kategoriaan tai samoihin kategorioihin, he kykenevät kommunikoimaan toistensa kanssa. Kommunikointi mahdollistaa oppimisen epäsuoran kokemuksen kautta.

Käsitteelliset kombinaatiot. Käsitteet voivat toimia korkeamman asteen käsitteiden rakennuspalikoina. Käsitteiden yhdistely mahdollistaa aivan uusien käsitteiden ilmaisemisen ja tulevaisuuden luovan visioinnin.

Viimeisen 20 vuoden aikana tehdyt empiiriset tutkimukset käsitteiden muodostuksesta osoittavat [46], että sisällytämme suhteellisen monimutkaista informaatiota käsitteeseen määrittellessämme sen piirteitä. Kokeet osoittivat myös, että käsitteillä ja käsitesysteemeillä on rikas sisäinen rakenne ja että kategorisointi riippuu sekä käsitteen muodostuksen tavoitteesta että sen kontekstista.

Yhteenvetona Wrobel toteaa [46], että käsitteet syntyvät tarkoituksellisen toiminnan tuloksena. Oliot ryhmitellään yhteen, koska ne pelaavat samaa roolia jossakin erityisessä kontekstissa. Tämä ryhmittely muunnetaan käsitteeksi erillisessä abstrahointi- tai käsitteellistämisprosessissa.

Wrobel katsoo, että käsitteiden ohjelmallinen käsittely vaatii kuvaussysteemiltä valintoja sekä esitysformalismin että esityskielen tasolla [46]. Esitysformalismilla tarkoitetaan

tässä esityksen syntaksia ja semantiikkaa. Esityskielellä tarkoitetaan kielioppia ja sanastoa, joka tehdään saatavaksi valitussa formalismissa. Wrobel korostaa, että sanasto ei tarkoita syntaktisten termien joukkoa vaan käsitteiden systeemiä. Hän katsoo, että saatavilla olevat psykologiset tosiasiat ihmisen käyttämistä käsitteistä ja niiden muodostumisesta mahdollistavat kuvaussysteemin edellyttämien vaatimusten esittämisen kummallakin tasolla. Valitun esityksformalismin pitää pystyä ilmaisemaan sellaiset käsitelmäritelmät, jotka psykologit ovat nähneet välttämättömiksi ihmisen käsitteiden selvittämisessä. Kielen tasolla ratkaistaan, mitkä käsitteet pitäisi sisällyttää esitykseen ja mitkä jätetään esityksen ulkopuolelle.

3.2 Käsite tietojenkäsittelyssä

Automaattinen tietojenkäsittely edellyttää erityisesti kognitiivisen ekonomian ja pakollisen jaettavuuden periaatteiden noudattamista tietojärjestelmissä käytettävien käsitteiden valinnassa. Meidän tulisi pystyä kuvaamaan kohdealue yhtenäisillä, käyttökelpoisilla, mahdollisimman laajaan tulevien käyttäjien yhteiseen sopimukseen ja tulkintaan perustuvilla käsite rakenteilla. Tietojärjestelmien suunnittelijat eivät läheskään aina ole kohdealueen asiantuntijoita. Mikäli kohdealueen tuntijoita ei ole mukana järjestelmään toteuttavien käsitteiden valinnassa ja määrittelyssä, on ilmeisenä vaarana, että järjestelmät tulevat toteutetuiksi käsitteillä, jotka vastaavat huonosti kohdealuetta koskevaa tietämystä ja kohdealueen toimijoiden tarpeita. Tietosisältöjen lisäksi tämä heijastuu käyttöliittymän logiikkaan, tietojen löytymiseen, järjestelmien integrointiin sekä rakennettavan informaatioresurssin uudelleenkäytettävyyteen ja laajennettavuuteen.

Tietosysteemejä suunnitellessamme ei ole yhdentekevää se, miten itse tajuamme käsite-termin merkityksen tai miellämmekö esimerkiksi tieto- ja käsitetermit toistensa synonyymeiksi. Seuraavassa esitettävät määritelmät on esitetty Tampereen yliopiston Käsitteellisen mallintamisen kurssilla [24].

Käsite on kohdealuetta koskevasta tietämyksestä muodostuva rakenteellinen kokonaisuus. Sen symbolia kutsutaan *termiksi*. Termin merkitys on se käsite, jota termi edustaa. *Käsitteen merkitys* on se kuvattavan maailman ilmiö, jota käsitteen avulla halutaan ilmaista. Käsitteiden välillä voi olla useita erilaisia *suhteita* [25]:

1. Käsitteet a ja b voivat olla toisistaan täysin *erillisiä*, ts. niissä ei ole yhtään yhteistä tunnusmerkkiä.
2. Käsitteillä a ja b voi olla yhteisiä tunnusmerkkejä, ts. ne ovat *tunnusmerkkiyhteisiä*.

3. Käsite b voi sisältyä käsitteeseen a, ts. kaikki käsitteen b tunnusmerkit ovat samalla myös käsitteen a tunnusmerkkejä. Tällöin sanomme, että käsite a *sisältää intensionaalisesti* käsitteen b.

Kaupin mukaan [25] minkä tahansa kahden käsitteen väliin voidaan aina määritellä uusi käsite. Tämä viittaa käsiteavaruuden äärettömyyteen. Yhteisten sanastojen määrittelyn tarkoituksena on saada meidät käyttämään tietojenkäsittelyssä käsitteitä, joiden rakenteesta olemme yhdessä sopineet. Sanastot ovat käsiteavaruuden äärellisiä osajoukkoja.

Käsiterakenne on käsitteen määritelmähierarkian kuvaus. Se koostuu määriteltävän käsitteen nimestä, määrittelevistä käsitteistä, kullakin hierarkiatasolla määriteltävän käsitteen ja määrittelevien käsitteiden välisistä intensionaalisista sisältymissuhteista sekä erilaisista rajoituksista, ehdoista ja muista lisämääreistä. Käsiterakenteen graafisen kuvauksen yleinen muoto on intensionaaliseen sisältymissuhteeseen perustuva suunnattu sykliton verkko, jota voidaan täydentää lisämerkinnöillä. Käsiterakenne esittää käsitteen määrittelyn.

Käsitekaavio (conceptual schema) on kohdealueen käsitteellinen kuvaus. Se ei ole kuvaus tietokannassa olevista tiedoista [20]. Käsitekaavio yhdistää tietojärjestelmiä. Käsitekaavio määrittelee sen, miten asiat liittyvät toisiinsa. Sen jostakin osasta tehdään mahdollisesti tietokanta ja osa käsitekaaviosta sisältyy kohdesysteemiä kuvaaviin ohjelmiin ja hakemistoihin. Käsitekaavion pitää sisältää kaikki kohdealueen relevantit käsitteet ja säännöt. Käsitteellistämisperiaatteen mukaisesti käsitekaaviossa kuvataan vain kohdealueen käsitteellisesti oleellisia piirteitä. Käsiterakenteina kuvatut käsitteet muodostavat käsitejoukon, johon sisältyvillä käsitteillä kohdealueen käsitteellinen malli muodostetaan. Määrittelyistä käsitteistä ja niiden välisistä suhteista syntyy abstrakti rakenne, joka muodostaa kohdesysteemiä ja sen käyttäytymistä kuvaavan teorian. Tämä edellyttää kuitenkin, että mallin sisältämät käsiterakenteet ja niiden suhteet ovat loogisesti ristiriidattomia.

Käsitteen määritelmä on kielellinen rakennusohje, jonka mukaisesti käsitteeseen sisältyvä tietämys kootaan tai muodostetaan tietämysalkioista ja/tai muista käsitteistä. Käsitteen määrittelytyyppinä on periaatteessa lukematon määrä, mutta käytännössä käytetään lähinnä seuraavaa määrittelytyyppiryhmäluokittelua:

1. Käsitteen *analyttinen eli intensionaalinen määrittely*, jossa käsitteen tunnusmerkit, ts. siihen intensionaalisesti sisältyvät toiset käsitteet ja niiden väliset suhteet ja muut tietämysalkiot, kuvataan jotain soveltuvaa määrittelytekniikkaa käyttäen.

2. Käsitteen määrittely käsitteen *ulkoisten suhteiden avulla*, jossa spesifoidaan määriteltävän käsitteen suhteet niihin saman käsitejärjestelmän muihin käsitteisiin, jotka eivät sisälly määriteltävään käsitteeseen. Esimerkkinä ulkoisten suhteiden käytöstä on aksiomaattinen määrittelytapa, jossa määritellään aksiomia käyttäen käsitteen suhteet ainakin joihinkin muihin järjestelmän käsitteisiin, mutta ei esitetä käsitteen intension analyysia.

3. *Representationaalinen eli esityksen määrittely*, jossa määrittely ilmaistaan spesifioimalla se tapa, miten määriteltävää käsitettä edustava arvo johdetaan määritteleviä käsitteitä edustavista arvoista. Kangassalo toteaa [24], että esityksen määrittely on lähinnä nettelytapa, jolla on huomattava käytännöllinen merkitys tietosysteemien kuvaamisessa ja suunnittelussa.

Kangassalo huomauttaa [24], että käsitteen käytön kannalta kuhunkin ryhmään kuuluvan määritelmän antama tietämys on yksinään puutteellista, joten tietämyksen kattavuuden pyrkivän määritelmän on sisällettävä ainakin em. ryhmiin kuuluvia osia.

Käsitteen sisältö on sen *intensio*. Se muodostuu muista käsitteistä, tietämysprimitiiveistä ja siitä rakenteesta, jonka ne muodostavat käsitteessä. Jos käsitteellinen malli on käsite, joka määrittelee koko kohdealueen, niin mallin koko käsitteellinen sisältö on saman käsitteen intensio. Käsitteitä, tietämysprimitiivejä ja rakennetta, jonka ne muodostavat käsitteen intensiossa, kutsutaan käsitteen *ominaispiirteiksi* eli tunnusmerkeiksi.

Niiden objektien joukkoa (kuin myös dataa, jota nämä kohteet edustavat), johon käsite on sovellettavissa, kutsutaan käsitteen *ekstensioksi*. Käsitteellä on aina intensio, mutta sen ekstensio voi olla tilapäisesti tai aina tyhjä. Ekstension alkioita kutsutaan käsitteen ilmentymiksi.

Mikäli käsitteen määrittelyssä tarvittavat asiat ja kohteet kyetään rajaamaan riittävällä tarkkuudella, voidaan käsiterakenne laatia mitä tahansa asiaa tai ilmiötä koskevan käsitteen kuvaamiseksi.

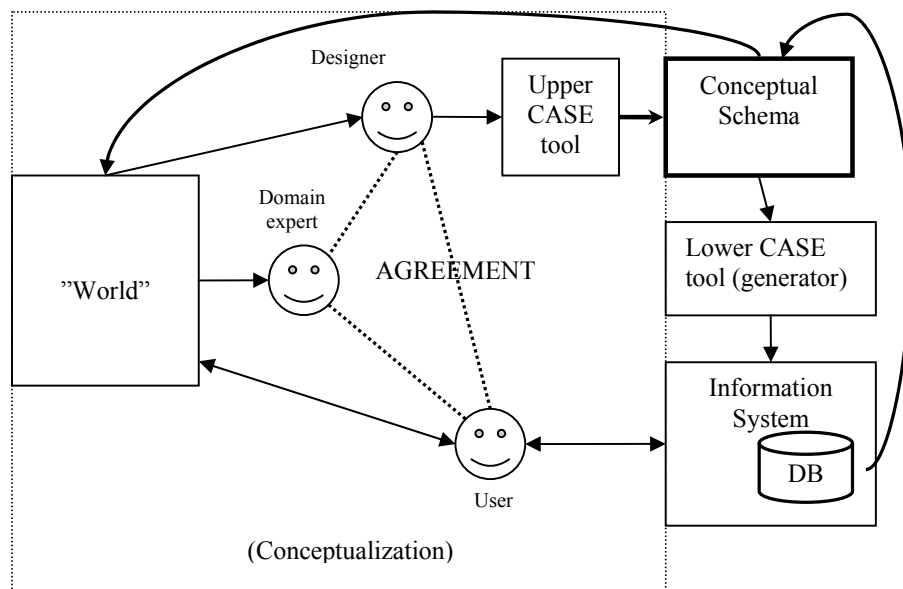
Intensio- ja ekstensio-termejä käytetään myös edellisestä poikkeavasti alan kirjallisuudessa. Esimerkiksi Elmasrin mukaan relaatiotietokannan kaavaa kutsutaan joskus myös intensioksi ja tietokannan tilaa ekstensioksi [7].

3.3 Yhteenveto

Se, miten me käsitteellistämme ympärillämme olevan maailman ja siihen sisältyvät osa-alueet, määrää käyttämämme käsitteiden merkityksen. Käsitteellistämisen tulos riippuu ympäröivästä kulttuurista, käsitettä muodostavan henkilön tietämyksen määrästä havainnoitavasta kohteesta sekä henkilön yksilöllisistä havaintomekanismeista.

Kognitiotieteissä käsitteen intension määritellään sisältävän sekä kategorian jäsenyyden määrittelevät kriteerit että ne ominaisuudet, jotka määrittelevät käsitteen suhteet toisiin käsitteisiin. Käsitteen ekstensio määritellään joukoksi olioita, joihin käsite viittaa [9]. Käsitteellisessä mallintamisessa käsitteen intensio määritellään käsitteen sisällöksi, joka muodostuu käsitteen sisältämistä muista käsitteistä, tietämysprimitiiveistä sekä edellisten muodostamasta rakenteesta käsitteen määritelmähierarkiassa. Ekstensioksi sanotaan niiden kohteiden joukkoa, johon käsitettä sovelletaan. Ekstensioiden määritelmien voidaan katsoa olevan yhteneväiset. Mikäli kognitiotieteissä tarkoitetut kategorian jäsenyyden määrittelevät kriteerit sekä käsitteen suhteita muihin käsitteisiin määrittelevät ominaisuudet tulkitaan käsitteellisessä mallintamisessa tarkoitetuiksi ominaispiirteiksi, määritelmät vastaavat toisiinsa myös intension tasolla.

Kuva 1. Kohdealueen käsitteellistäminen ja tietosysteemi [29]



Kuvan 1 kaavio on mukaelma Meersmanin esityksestä [29]. Kaavio esittää yhteisöllisen käsitteellistämisen ja tietojärjestelmän välistä vastaavuutta. Sekä kohdealueen asiantuntijoilla (domain expert), tietojärjestelmän käyttäjillä (user) että tietokannan pohjana

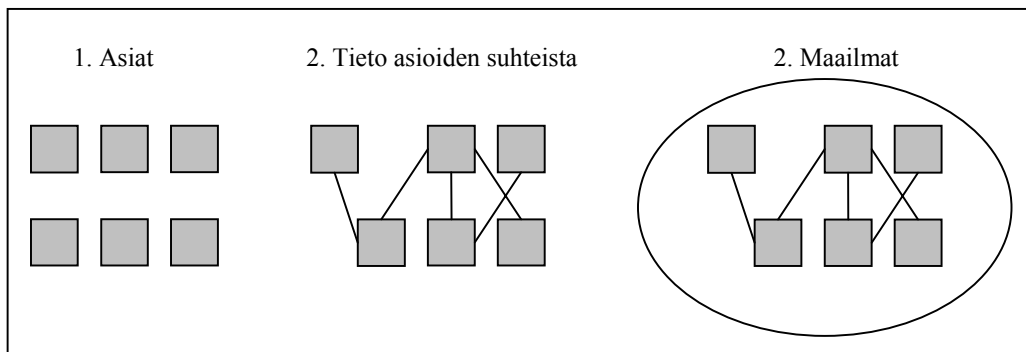
tijoilla (domain expert), tietojärjestelmän käyttäjillä (user) että tietokannan pohjana olevan käsitekaavion rakentajalla pitää olla yhtenevä tulkinta (agreement) mallinnettavasta maailmasta (world), ts. järjestelmä on suunniteltava toteuttamaan yhdessä sovitut käsitekenteet. Kuvassa 1 vaaleanharmaalla esitetyn suorakaiteen sisälle jäävä osio kuvaa käsitteellisen mallintamisen prosessia, jota käsittelem tarkemmin tämän työn kohdassa 4.2. Upper CASE tool voisi olla esimerkiksi Concept-D -kieli ja Lower CASE tool vaikkapa jokin XML-editori. Concept-D kielen perusteet esittelen tämän työn kohdassa 5. Nuoli informaatiojärjestelmän tietokantaosasta (DB) käsitekaavioon kuvaa sitä, että osa käsitekaaviosta toteutetaan järjestelmän tietokannassa.

4. Käsitteellinen mallintaminen

Käsitteellinen mallintaminen on laaja alue, jonka voidaan katsoa sisältävän mm. eri kohdealueiden käsite- ja ontologiakirjastojen suunnittelun, tarvittavien systeemien suunnittelun ja erilaisten metatietojenkin määrittelyn. Vaikka käsitteellä itsellään ei ole semantiikkaa, vaan sisältö, semantiikasta puhutaan kaikkien edellisten yhteydessä. Tässä luvussa tarkastelen aluksi semantiikka-termin määritelmää ja termin käyttöä käsitteellisen mallintamisen yhteydessä. Kohdassa 4.1 esittelen ontologioista, sanastoista ja metatiedoista käytyä keskustelua ja kohdassa 4.2 kuvaan mallintamisprosessin eri vaiheet.

Semantiikka tarkastelee merkkien esittämää tehtävää, niiden viittaussuhteita merkkijärjestelmän ulkopuolelle, merkkijonojen tulkintaa ja merkityksiä. Semantiikalla tarkoitetaan myös yleisesti merkitysteoriaa; joskus myös teoriaa jonkin yksittäisen kielen sanojen (ja lauseiden) merkityksistä [3, 36]. Semantiikalla voidaan tarkoittaa myös resurssien tulkintaa, merkityksiä ja viittauksia toisiin resursseihin [20].

Kuva 2. Semanttisten sovellusten kolmijako Dacontan mukaan [20]



Semantiikan vahvuus ilmenee käytettävissä olevasta ilmaisun tai kuvailun rikkauden tasosta (tyhjentyvyydestä). Heikoilla semanttisilla malleilla kyetään ilmaisemaan vain hyvin yksinkertaisia merkityksiä. Semanttiset sovellukset voidaan edelleen jakaa kolmeen tasoon [20]: (Asiaa havainnollistaa kuva 2.)

- asioita mallintaviin (esim. XML Schema),
- asioiden suhteita ja toimintatapoja mallintaviin (esim. RDF) ja
- asiakokonaisuuksia (maailmoja) mallintaviin sovelluksiin (ontologiat).

Semanttinen informaatio kuvaa Kangassalon mukaan tietokannassa tai tietosysteemin tiedostoissa olevien tietojen asiasisällöllisen merkityksen sekä ne pysyväisluonteiset rajoi-

tukset, joita tietojoukon tietojen käsittelyssä on tietojen merkityksen virheettömänä säilymisen vuoksi noudatettava [22]. Kangassalo esitti jo 1980-luvun lopulla yhdeksi semanttisen informaation käytön syyksi pyrkimyksen kehittää menetelmiä ja välineitä, joiden avulla voidaan etsiä ja käyttää halutun merkityksen omaavia tietoja.

Kangassalon mukaan [22] tiedonhallinnan alueella semanttisesta informaatiosta puhutaan käsitekaavioiden (conceptual schema) ja semanttisten tietomallien yhteydessä (semantic data model). Tekoälyn alueella siitä puhutaan tietämuskantojen ja tietämyksen esittämisen (knowledge representation) yhteydessä. Ohjelmoinnin alueella semanttinen informaatio liitetään abstrahoinnin (abstraction) eri muotoihin. Kangassalo käyttää kaikista tietokantojen ja tietosysteemien tietojen semantiikan suunnitteluun ja kuvaamiseen liittyvistä toimenpiteistä yhteisnimitystä käsitteellinen mallintaminen (conceptual modeling).

Kangassalo määrittelee [22] käsitteellisen mallintamisen prosessiksi, jossa

- luodaan, tunnistetaan ja löydetään informaation kohdealuetta kuvaavat relevantit käsitteet ja käsitteelliset mallit sekä
- rakennetaan tai valitaan informaation se relevantti käsitteellinen sisältö, joka on tarkoitus sisällyttää kohdealuetta kuvaavaan informaatioon.

Hän toteaa tietosysteemin suunnittelun edellyttävän, että käsitteet ja säännöt (so. kohdealueen käsitekaavio) tehdään näkyväksi ja jaettavaksi systeemin käyttäjien kesken. Hänen mukaansa kohdealueen graafinen käsitekaavio, joka tukee datan käsittelyä, voidaan nähdä itse tietosysteemin kuvauksena. Tiedon rakenteen ja merkityksen tehokas hallinta edellyttävät, että käytettyjen käsitteiden sisältö tunnetaan riittävän hyvin [21].

Kangassalon mukaan tietosysteemien esittämiseen käsitteellisinä spesifikaationa kohdistuva tutkimus juontaa juurensa 1950-luvulle, jolloin Young ja Kent kehittivät abstraktin mallin tietojärjestelmien käsitteellisestä kuvauksesta. Senko esitti 1970-luvulla FORAL LP -systeemin, jossa korkean tason graafinen käyttöliittymä perustui verkkorakenteisen tiedonhallintajärjestelmän päälle rakennettuihin binäärirelaatioihin [21]. Alunperin käsitteellistä kaaviota käytettiin kuvaamaan tietokantaan talletettujen tietojen semantiikkaa. Sittemmin käsitys alkoi muotoutua suuntaan, jonka mukaan käsitteellinen kaavio ensisijaisesti kuvaa kohdealuetta ja vain toissijaisesti selittää datan merkityksiä [21].

Käsitteellinen mallintaminen käsitteenä ei ole ristiriidaton tutkijoiden keskuudessa. Selvimmin tutkijat jakautuvat kahteen koulukuntaan, joista toinen tukeutuu käsitteiden ilmentymien (ekstension) ja toinen käsitteiden sisäisen rakenteen (intension) tutkimiseen.

Calvanese ja muut [4] toteavat, että sovellusalueen käsitteellinen mallintaminen ja sen käsitteellisen esittämisen avulla mahdollistuva päättely ovat kriittisessä asemassa informaation integrointia vaativien sovellusten suunnittelussa ja ylläpidossa. He esittävät integroinnin yleistä arkkitehtuurimallia, joka perustuu eksplisiittisesti metadataan perustuvaan sovelluksen käsitteelliseen esittämiseen. Arkkitehtuurimalli sisältää metatason, käsitteellisen tason, loogisen tason ja fyysisen tason. Mallin käsitteellisen tason kuvaus on riippumaton systeemin toteutuksesta ja mallin tarkoituksena on sovelluksen semantiikan esittäminen.

Ekstensionaalaisella mallintamistavalla tarkoitetaan käsitteen ilmentymäjoukkojen analysointiin perustuvaa mallintamista; esimerkiksi ER (Entity Relationship)-mallit [7] ovat ekstensionaaliseen kuvaukseen perustuvia malleja. ER-malleista puhutaan usein myös semanttisina malleina. Calvanesen artikkelin tarkempi tutkiminen osoittaa hänen tarkoittavan käsitteellisellä mallintamisella joukko-oppiin perustuvaa, ekstensionaalista lähestymistapaa.

Kangassalon mukaan ER-mallin mukainen lähestymistapa sopii hyvin teolliseen ympäristöön, jossa kohdealue muodostuu selkeästi identifioitavista konkreettisista kohteista ja niiden välisistä suhteista [21]. ER-mallin mukaisessa lähestymistavassa mallinnettavat käsitteet luokitellaan kolmeen kategoriaan: kohteisiin, suhteisiin ja attribuutteihin. Ne siis muodostavat ER-mallin ontologian. Suunnittelija saattaa joutua tekemään työn alkuvaiheessa ratkaisuja, joiden perusteluiksi hänen tietomääränsä kohdealueesta eivät vielä riitä. Kangassalon mukaan ER-mallin mukainen luokittelu ei ole ominaista tietämyksen sisäisille ominaispiirteille, koska se pakottaa ne ennalta määrättyyn rakenteeseen [21].

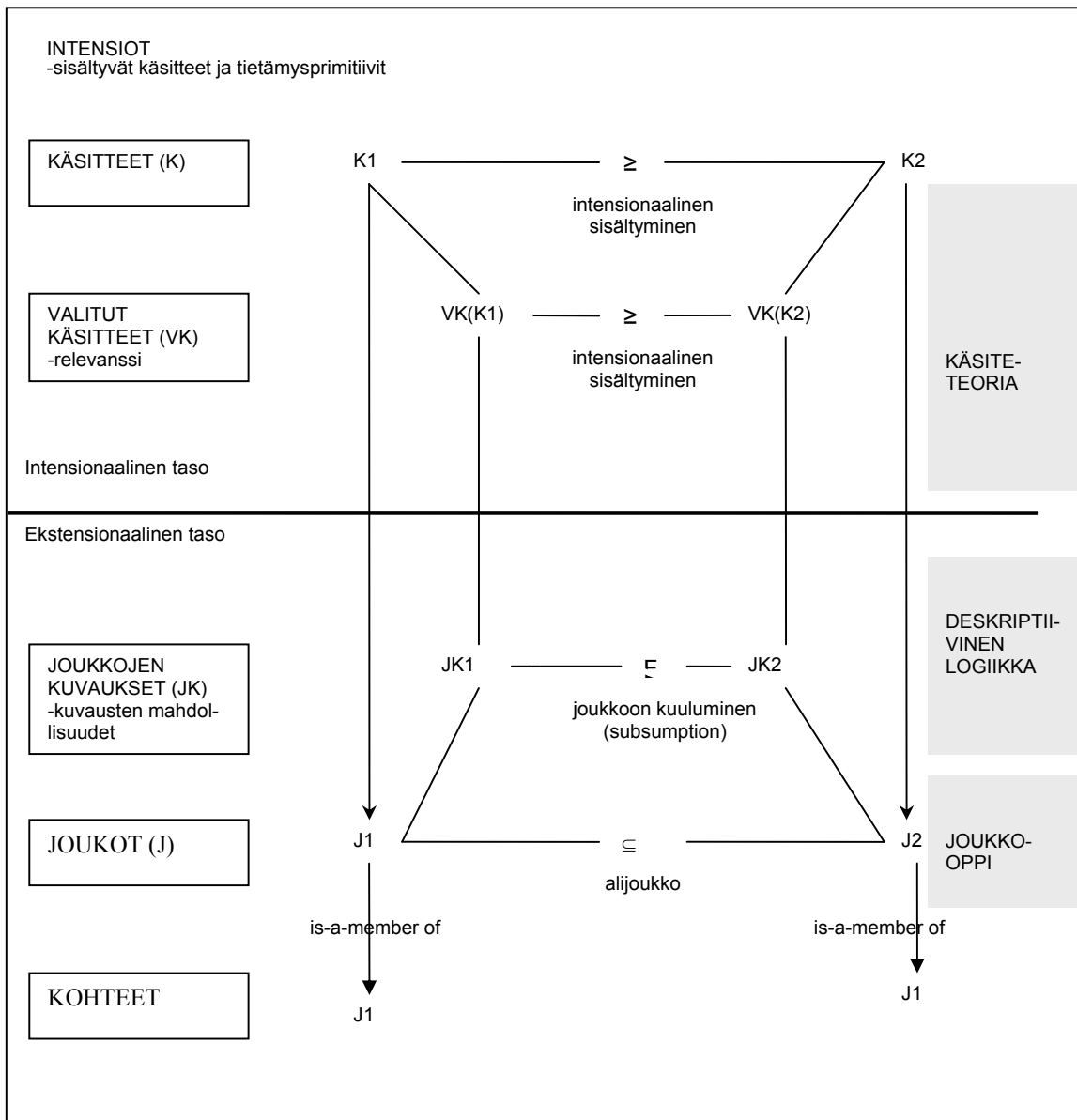
Järjestelmäintegroinnissa törmätään ongelmiin, jotka aiheutuvat mm. kohdealueen yksittäisten osien hallintaan rakennettujen järjestelmien toisistaan poikkeavasta tavasta nimenä käsitteitä edustavia termejä tai luokitella jokin yhteinen käsite toisessa järjestelmässä suhteeksi ja toisessa vaikkapa attribuutiksi.

Kangassalon mukaan käsitteen muodostus ja käsitteellinen mallintaminen perustuvat sisällöllisiin aspekteihin, jotka alkavat muodostua jo paljon aikaisemmin kuin meillä on vielä yhtään käsitteen ilmentymää tai edes hypoteesia siitä. Intensionaalinen lähestymistapa painottaa käsitteiden sisältöä ja rakennetta, jota käytetään mallintamaan ilmiötä todellisessa

tai hypoteettisessa maailmassa [23]. Kangassalo korostaa, että intensionaalisessa mallintamisessa kuvataan jonkin määrätyn käsitteen sisältöä, kun taas ekstensionaalisessa mallintamisessa kuvataan joukkoja [24].

Kuvan 3 esittämässä kaaviossa mallintamisen intensiotaso ja ekstensiotaso on erotettu selkeästi toisistaan. Intensiotasolla tutkitaan käsitteiden sisäistä rakennetta ts. niihin sisältyviä toisia käsitteitä ja tietämysprimitiivejä. Käsitteen käytön tarkoitus määrittää ne ominaisuudet, jotka tarkoituksen kannalta relevantteihin valittuihin käsitteisiin sisällytetään. Tällä tasolla yleisempi käsite sisältyy erikoistuneempaan, ts. enemmän ominaisuuksia sisältävään käsitteeseen. Olkoon K1 työntekijä-käsite ja K2 henkilö-käsite. Intensionaalisella tasolla K1 sisältää käsitteen K2, ts. työntekijä-käsite sisältää henkilö-käsitteen. Asia tuntuu aika selvältä; sisältyyhän jokaiseen työntekijään se ominaisuus, että hän on henkilö. Mikäli määriteltävään käsitteeseen sisältyy käsitteitä, joilla on määrittelevinä käsitteinä jokin yhteinen käsite, käsite rakenne muodostaa suunnatun syklittömän verkon, Mikäli näin ei ole, rakenne pelkistyy puuksi.

Kuva 3. Käsiterakenteet, joukkojen kuvaukset ja joukkojen ilmentymät [24]



Kun käsitteet on valittu, analysoitu ja määritelty, on tiedossa myös niiden ominaispiirteet. Tällöin voidaan alkaa tutkia valittujen käsitteiden keskinäisiä suhteita, ts. muodostamaan kohdealueen ontologista analyysia. Tällä tasolla työntekijäjoukko (JK1) sisältyy henkilöiden joukkoon (JK2). Kuvaukset voidaan tehdä deskriptiivisellä eli kuvailevalla logiikalla. Deskriptiiviset logiikat ovat tietämyksen esittämiskieliä, joita voidaan käyttää jonkin sovellusalueen tietämyksen esittämiseen rakenteisella, formaalilla ja ymmärrettävällä tavalla. Guarino et al:n mukaan [17] sisältyvyysuhde (subsumption) (aliluokka, is-a, jne.) muodostaa luokituksen perustan. Sisältyvyysuhde on tärkeä työkalu muodostettaessa onto-

logian rakennetta. Kaikki määrätyn aliluokan ilmentymät ovat välttämättä myös tämän ylikuokan ilmentymiä.

Kangassalo huomauttaa [24], että käsitteet ovat abstrakteja rakenteita, joita voidaan käyttää kuvauksina mm. tietämyksen organisoinnissa, käsitejärjestelmien rakentamisessa, tieteellisten teorioiden teossa, ontologioiden rakentamisessa, taksonomioitten teossa, kirjasto- luokituksissa, tietokantojen suunnittelussa, ohjelmistojen jäsentämisessä, tietojärjestelmien suunnittelussa ja ymmärryksen lisäämisessä kohdealueesta.

4.1 Ontologioista, sanastoista ja metatiedoista

Filosofian ontologia tutkii kohteiden, ominaisuuksien, tapahtumien, prosessien ja suhteiden rakenteita ja lajeja todellisuuden kaikilla alueilla. Ontologia-termiä käytetään usein metafysiikka-termin synonyyminä. Smithin mukaan Aristoteles itse käytti ilmaisua 'ensimmäinen filosofia', jota hänen varhaiset seuraajansa alkoivat nimittää metafysiikaksi. Ontologia-termiä tiedetään käyttäneen toisistaan tietämättä 1600-luvulla sekä Rudolf Göckel että Jacob Lorhard. Termi ilmestyi englantilaiseen Baileyn Dictionaryyn 1700-luvulla [35].

Smithin mukaan Aristoteleen koulukuntaa edustavien filosofien luokitukset ovat erilaisesta yleisyystasostaan huolimatta monella tavalla verrattavissa luonnontieteiden, erityisesti biologian ja kemian luokituksiin. Muista filosofisista koulukunnista poiketen aristotelaiset pyrkivät löytämään perusluokituksia asioille ja ilmiöille, eivät niinkään näiden selityksiä.

Ontologiat tietojenkäsittelyssä

Guarino et al:n mukaan yleisesti tietojenkäsittelyssä hyväksytty ontologia-termin merkitys [16] on jokseenkin riippumaton filosofisista edeltäjistään ja miltei samaistuu termiin käsitteellinen malli. Heidän mukaansa kohdealueen käsitteellinen malli on kohdealueen ontologian aktueelli toteutus, jonka on toteutettava ajettavan ohjelmiston tuotannolliset tarpeet. Ontologian suunnittelu on riippumaton ajonaikaisesta toteutuksesta ja sen ainoa tarkoitus on spesifioida sovelluksen kohdealueen käsitteet [15]. Samaa mieltä on Kangassalo, joka toteaa, että ontologia ei ole sovellus- tai järjestelmäkohtainen. Ontologia määrittelee peruskäsitteet, joista voidaan rakentaa käsitekaavio [24].

Guarinon mukaan jokaisella informaatiosteemillä on oma ontologiansa, jossa kuvataan systeemissä käytettyjen symbolien merkitys. On olemassa hyviä ontologioita, jotka perustuvat käsitteiden semanttisiin merkityksiin ja huonoja ontologioita, joissa kohdealueen käsitteet ja ilmiöt on luokiteltu niin karkealla tasolla, että niiden käyttö informaatiosteemien suunnittelussa ja niiden pohjalta toteutetuissa systeemeissä johtaa ylläpidon ja uudelleenikäytön ongelmiin [14].

Noy et al. [31] määrittelevät ontologian yhteiseksi sanastoksi, joka sisältää kohdealueen peruskäsitteiden ja niiden suhteiden määritelmät koneen tulkittavissa olevassa muodossa. Heidän mukaansa ontologia on formaali, eksplisiittinen kuvaus [31] kohdealueen käsitteistä, niiden erilaisista piirteistä ja attribuuteista koostuvista ominaisuuksista, suhteista ja rooleista sekä näiden rajoituksista. He näkevät ontologian kehittämiseen mm. seuraavat syyt:

- ihmisten ja ohjelmistoagenttien jakaman informaation rakenteen yhteinen ymmärtäminen,
- kohdealueen tietämyksen uudelleenikäytön mahdollistaminen,
- kohdealuetta koskevien oletusten ja väittämien esittäminen yksikäsitteisesti,
- kohdealuetta koskevan tietämyksen erottaminen toiminnallisesta tietämyksestä,
- kohdealueen analysointi.

Noy et al.:n mukaan ontologia yhdessä sen edustamien käsitteiden yksilöllisten ilmentymien kanssa muodostaa tietämyskannan. He toteavat myös, että raja ontologian ja tietämyskannan välillä on todellisuudessa häviävän pieni.

Tiedon jakamista ja uudelleenikäyttöä tutkiva Gruber korostaa ontologian rakentamisen tarkoituksen tunnistamista ja määrittämistä ennen ontologian suunnittelun aloittamista [11]. Hän määrittelee ontologia-termin tarkoittavan tiedon jakamisen alueella käsitteellistämisen spesifikaatiota, joka on kuvaus niistä käsitteistä ja käsitteiden välisistä suhteista, joilla ohjelmistoagentit kommunikoivat keskenään; ts. joihin ne ovat sitoutuneet ontologisesti. Gruber toteaa, että käytännössä ontologinen sitoutuminen on sopimus käyttää jotakin sanastoa tavalla, joka on ehyt suhteessa jonkin ontologian spesifioimaan teoriaan.

Myös Chandrasekaran et al.:n mukaan [5] ontologiat ovat jossakin erityisessä tietämysalueessa olevien kohteiden ja niiden välisten suhteiden sisältöteorioita, jotka tuottavat ne potentiaaliset termit, joilla kuvaamme tietämystämme kyseessäolevasta kohdealueesta.

Vertaillen erilaisia ontologiaeditoreita WonderTools-projektissa [6] tutkijat ki-teyttivät tarkoituksiinsa parhaiten sopivan ontologian määritelmän seuraavasti: Ontologia on formaali, eksplisiittinen, jaetun käsitteellistämisen spesifikaatio. Määritelmä voitaneen katsoa yhtäpitäväksi Gruberin [11] määritelmän kanssa. Käsitteellistämällä määritelmässä [6] tarkoitetaan tietyn maailman ilmiöiden abstraktia mallia, jossa ko. ilmiöiden relevantit käsitteet on identifioitu. Eksplisiittisyys viittaa siihen, että käytettyjen käsitteiden tyypit ja niiden käytön rajoitukset on tarkoin määritelty. Formaalisuus tarkoittaa sitä, että koneen pitää pystyä lukemaan ontologiaa; se on siis esitettävä jollakin muulla kuin luonnollisella kielellä. Jaettavuus tarkoittaa sitä, että käsitteellistäminen ei ole joidenkin yksilöiden omaa, vaan sen on hyväksynyt jokin suurempi ryhmä.

Guarino et al. ovat havainneet [14] ontologia-termille käytetyn tutkijoiden keskuudessa ainakin seitsemää erilaista tulkintaa:

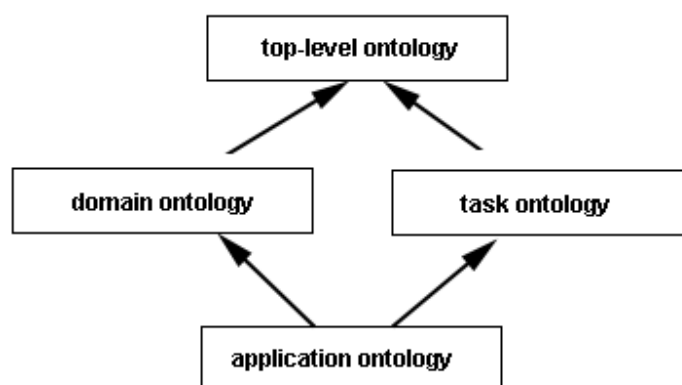
- ontologia filosofisena tieteenalana,
- ontologia käsitteellisenä tietosysteeminä,
- ontologia käsitteellistämisen spesifikaationa,
- ontologia formaalina semanttisena kuvauksena,
- ontologia loogisen teorian kautta muodostettuna käsitteellisenä systeemin esi-tyksenä,
 - jota luonnehtivat spesifit formaalit ominaisuudet
 - jota luonnehtii vain sen spesifi tarkoitus
- ontologia loogisen teorian käyttämänä sanastona,
- ontologia loogisen teorian metatason spesifikaationa.

Guarino et al. ehdottavat, että puhuttaessa ontologiasta filosofian haarana, käytettäisiin termiä Ontologia kirjoitettuna isolla alkukirjaimella erotuksena muista ontologioista, joiden tulkinnasta voidaan olla monta mieltä. He huomauttavat, että esimerkiksi edellä olevassa tulkintojen luokittelussa kolmantena esitetty tulkinta riippuu viime kädessä siitä, mitä tarkoitetaan spesifikaatiolla ja käsitteellistämällä.

Guarino et al. jakavat ontologiat kolmelle tasolle niiden kuvaaman käsitemaailman yleisyysasteen mukaan. Jaottelu on esitetty kuvassa 4. Ylätason (top-level) ontologia kuvaa

jostain erityisestä ongelmasta tai sovellusalueesta riippumattomia yleisiä käsitteitä ja niiden suhteita. Tällaisia ovat mm. käsitteet tila, aika, asia, kohde, tapahtuma, toiminta jne. Kohdealueontologiassa (domain ontology) kuvataan jonkin erityisalan käsitteiden suhteet (kuten lääketieteen, autojen jne). Tehtäväontologia (task ontology) määrittelee erityisen tehtävän tai aktiviteetin (esimerkiksi diagnosoinnin tai myynnin) käsitteet. Sovellusontologiassa (application ontology) kuvataan käsitteet, jotka riippuvat sekä sovellusalueesta että siihen liittyvästä erityistehtävästä. Nämä käsitteet vastaavat usein sovellusalueen olioiden sellaisia rooleja, joihin yhdistyy jokin tietty tehtäväontologiassa määritelty käsite. Hierarkiassa alemman tason käsitteet määritellään käyttäen ylemmän tason käsitteitä

Kuva 4. Ontologiatyyppejä [13].



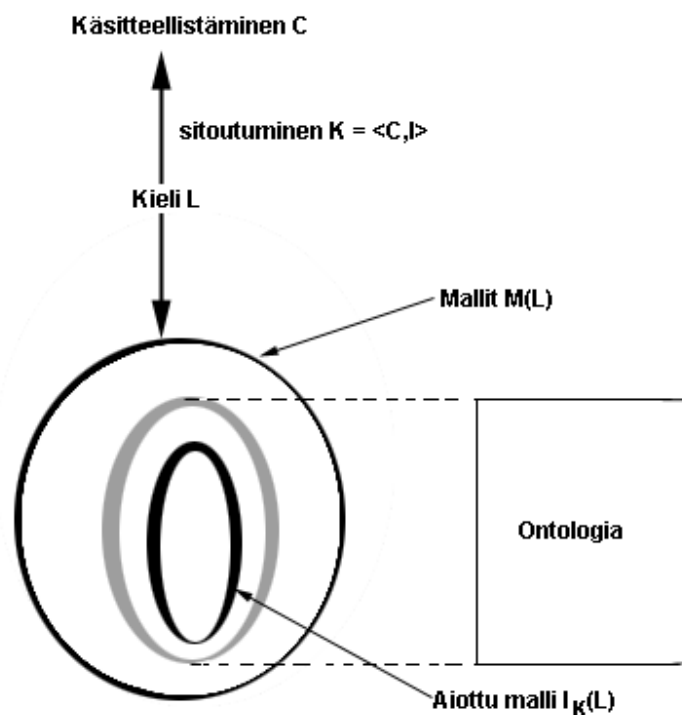
Sovellusontologia ja erityinen tietämuskanta yhdessä muodostavat yleisen tietämuskannan [14]. Erityisessä tietämuskannassa kuvataan tietyn toiminnan tilan tai tiedollisen tilan käsitteitä sekä niiden välisiä suhteita. Guarinon mukaan ontologia on riippuvainen käytetystä kielestä, kun taas käsitteellistäminen on kielestä riippumaton. Hän määrittelee, että kielellä L esitetty ontologia O approksimoi käsitteellistämistä C, jos on olemassa sellainen ontologinen sitoutuminen $K = \langle C, I \rangle$, että kielellä L esitettävät mallit ovat sopuolosuhteissa ontologian O malleihin sisältyvän ontologisen sitoutumisen K kanssa. Ontologia O sitoutuu käsitteellistämiseen C, jos

- se on suunniteltu kuvaamaan käsitteellistämisen C merkitystä ja
- se approksimoi käsitteellistämistä C.

Kieli L sitoutuu ontologiaan O, jos se sitoutuu johonkin sellaiseen käsitteellistämiseen C, että ontologia O on ristiriidaton käsitteellistämisen C kanssa. Tietämyksenhallintaan liittyen Guarino määrittelee ontologian loogiseksi teoriaksi, joka pitää kirjaa formaalin

sanaston tarkoitetuista merkityksistä, ts. se on ontologinen sitoutuminen johonkin tietyn maailman käsitteellistämiseen. Sellaista sanastoa käyttävät mallit rakentuvat ontologisen sitoutumisen perustalle. Ontologia heijastaa epäsuorasti tätä sitoutumista ja sen allaolevaa käsitteellistämistä approksimoimalla näitä tarkoitettuja malleja. Kuva 5 havainnollistaa määritelmää.

Kuva 5. Sanaston, käsitteellistämisen, ontologisen sitoutumisen ja ontologian väliset suhteet [13].



Mikäli ontologia on täsmällinen ja kohdealuettaan hyvin kuvaava, voidaan ontologiaa käyttää Guarinon mukaan tietojärjestelmän arkkitehtoonisena perustana, jolloin se ohjaa järjestelmän sovellusohjelmista, informaatioresursseista ja käyttöliittymistä muodostuvaa kokonaisuutta. Tällöin voidaan puhua ontologiaohjatuista informaatiojärjestelmistä [13]. Guarino tähdentää, että ontologian ylläolevassa määritelmässä tarkoitettu formaali sanasto ei välttämättä ole kielen osa, vaan se voi olla esimerkiksi protokolla, jolla ohjelmistoagentit kommunikoivat keskenään.

Laajoja ontologioita voidaan rakentaa integroimalla olemassaolevia eri kohdealueiden ontologioita ja ylemmän tason ontologioita. Mm. Lutzin mukaan ontologioiden uudelleen-

käytön ja integroinnin edellytyksenä on kuitenkin, että ne on esitetty täsmällisesti ja että ne perustuvat yleisesti hyväksytyihin ylätasen ontologioihin [28].

Määriteltäessä Webin ontologiakielen vaatimuksia työryhmä korosti, että kielellä pitää pystyä [42]

- ilmaisemaan ontologioita,
- muuttamaan julkaistuja ontologioita,
- integroimaan olemassa olevia ontologioita,
- ratkaisemaan ontologioiden ja niihin pohjautuvien käytännön toteutusten välisiä ristiriitaisuuksia,
- tuottamaan jonkinlainen tasapaino ontologioita luotaessa niiden ilmaisukyvyyn ja skaalautuvuuden välille,
- välttämään tarpeetonta, laajan käytön estävää kompleksisuutta,
- ylläpitämään yhteensopivuutta muiden standardien kanssa sekä
- tukemaan kansainvälisyyttä.

Kun W3C julkisti Webin ontologiakielen, Ontology Web Language (OWL), lehdistötiedotteessa todettiin mm., että OWL tarjoaa kielen, jonka avulla rakenteisten Web-ontologioiden määrittely on mahdollista. OWL mahdollistaa mm. sillä kuvattun ontologian hajauttamisen, skaalautuvuuden, yhteensopivuuden saavutettavuuteen ja kansainvälistymiseen liittyvien standardien kanssa sekä avoimuuden ja laajennattavuuden [45].

OntoClean

Guarino ja Welty ovat kehittäneet OntoClean-metodologian ontologioiden rakentamiseen ja analysointiin[16,17]. Se tuottaa heidän mukaansa välineet sekä ontologisille päätöksille että niiden arvioinnille. OntoClean-metodologia perustuu käsitteisiin, jotka ovat riittävän yleisiä käytettäväksi minkä tahansa, jostakin tietystä sovellusalueesta riippumattoman ontologian kehittämiseen. Näiden käsitteiden avulla määritellään joukko metaominaisuuksia, joita puolestaan käytetään kuvaamaan ontologian sisältämien ominaisuuksien, luokkien ja suhteiden relevantteja piirteitä. Metaominaisuudet tuottavat lisäksi ontologian luokittelurakenteelle joukon rajoituksia, joiden avulla voidaan arvioida tehtyjä valintoja. Näitä meta-

ominaisuuksia ovat oleellisuus (essence), joustamattomuus (rigidity), identiteetti (identity) ja ykseys (unity).

Jonkin kohteen ominaisuus on *oleellinen* (essential), jos se on kohteen ilmentymillä aina olemassa. Ominaisuus on *joustamaton* (rigid), jos se on oleellinen kaikille kohteen esiintymille. Esimerkiksi ominaisuutta 'olla henkilö' pidetään joustamattomana ominaisuutena. Ominaisuus on *puoli-joustamaton* (semi-rigid), jos se ei ole oleellinen kaikille kohteen ilmentymille. Ominaisuus on *joustava* (anti-rigid), jos se ei ole koskaan oleellinen.

Ontologiassa jokainen ominaisuus pitäisi luokitella joko joustamattomaksi, puoli-joustamattomaksi tai joustavaksi, koska näiden ominaisuuksien avulla testataan luokkasuhteiden ontologinen johdonmukaisuus. Esimerkiksi opiskelija-luokka ei voi sisältää henkilöluokkaa, koska opiskelija-ominaisuus on joustava ja henkilö-ominaisuus on joustamaton.

Identiteetti-ominaisuuden perusteella voidaan päätellä, ymmärretäänkö yksittäiset oliot samoiksi vai erillisiksi. Identiteetti-kriteerit ovat ehtoja, joita käytetään määrittelemään samuus (riittävät ehdot) ja jotka johtavat samuuteen (välttämättömät ehdot). Ykseys-ominaisuuden perusteella päätellään, hahmotetaanko määrätyistä osista muodostuva kokonaisuus yksilölliseksi olioksi siten, että tiedämme yleisesti, mikä osa kuuluu kohteeseen ja mikä ei kuulu siihen.

OntoClean-metodologian perusajatus on, että ontologiaa kehitettäessä kuhunkin sen ominaisuusluokkaan liitetään luokan käyttäytymisen suhteessa edellä kuvattuihin ontologisiin käsitteisiin (notions) kuvaavat metaominaisuudet. Kutakin ominaisuutta analysoidessaan mallintajalle aukeaa tärkeitä näkymiä esitettävän ontologian rakenteeseen ja luonteeseen.

Guarino et al.:n mukaan perusluokituksen (backbone taxonomy) tunnistaminen on eräs tärkeimmistä analyysin hyödyistä. *Kohdealueen perusluokitus* koostuu kaikista sisältyvyysuhteen mukaisesti organisoituneista joustamattomista ominaisuuksista. Se esittää käsiteltävän kohdealueen kaikki tärkeimmät ominaisuudet ts. se muodostaa vakaiden ominaisuuksien perusjoukon käsiteltävässä kohdealueessa. Tämä johtuu Guarinon ja Weltyin mukaan siitä, että jokaisella oliolla täytyy olla identiteetti-kriteerit ja jokin joustamaton ominaisuus, joka kuvaa nuo kriteerit. Perusluokitus nopeuttaa ontologioiden integrointiprosessia, koska jokaisen olion pitää edustaa ainakin yhtä luokituksen sisältävää ominaisuutta. Koska perusominaisuudet esittävät kohdealueen muuttumattomia, oleellisia piirteitä, on

niiden analysointi ensimmäisenä tärkeää. Kahden ontologian integrointi voidaan aloittaa vertaamalla niiden joustamattomia ominaisuuksia ilmaisevaa yleistä perusluokitusta.

Sisältyvyysuhde (subsumption) (aliluokka, is-a, jne.) muodostaa luokituksen perustan. Sisältyvyysuhde on tärkeä työkalu muodostettaessa ontologian rakennetta. Kaikki määrätyn aliluokan ilmentymät ovat välttämättä myös tämän yliluokan ilmentymiä. Sisältyvyysuhde on yleisimmin käytetty luokitusten rakenteellinen primitiivi. Hakemistojen ja oliokielten luokkahierarkioita muodostettaessa sisältyvyysuhde on ainoa mahdollinen luokittelukriteeri [17]. Guarino et al.:n mukaan sitä ei kuitenkaan ymmärretä aina oikein. Useimmiten virheitä tehdään, kun sisältyvyysuhde sekoitetaan 'olla-osa'-suhteeseen tai jonkin luokan ilmentymä ymmärretään virheellisesti luokan aliluokaksi.

Vaikka ontologioiden suunnittelijat perustavat joitakin metodeja olioperustaisen ohjelmistosuunnittelun teorioihin, he korostavat Noyn mukaan [31] ontologian kehittämisen eroja suhteessa olioparadigman mukaiseen luokkien ja niiden suhteiden määrittelyyn. Oliohjelmoinnin ratkaisujen perustana ovat ensisijaisesti jonkin luokan toiminnalliset ominaisuudet, kun taas ontologian ratkaisut perustuvat jonkin luokan rakenteellisiin ominaisuuksiin.

Metatiedoista ja sanastoista

Kreikankielistä termiä 'meta' vastaavat suomenkieliset sanat jälkeen ja välissä. Meta-tieto on määritelty pitkään pelkästään tiedoksi tiedosta. Yritysmaailmasta on peräisin määritelmä, jonka mukaan metatietoa on kaikki se fyysinen data ja tietämystä sisältävä informaatio organisaatiosta ja sen ulkopuolelta, jota organisaatio käyttää [26]. Tähän kuuluvat edellisen määritelmän mukaan mm. itse data, liiketoiminnallisten ja teknisten prosessien kuvaus, säännöt ja eheystarkistukset sekä rakenteet.

Osana valtion tietotekniikan rajapintasuosituksia [38] on määritelty sähköisen asioinnin tavoitearkkitehtuuri. Se muodostuu neljästä loogisesta kokonaisuudesta: käyttäjäliittymästä, tukipalveluista, metatiedoista ja perusjärjestelmistä (kuva 17). Metatiedoilla tarkoitetaan suosituksessa palvelukuvauksia sisältävän palveluhakemiston ja sanastokuvauksia sisältävän tietohakemiston muodostamaa loogista kokonaisuutta. Suosituksessa todetaan edelleen, että metatiedoilla tarkoitetaan myös tekniikoita, joilla voidaan kuvata tietomalleja, sisältöjä ja asiakirjoja.

Suosituksen [38] mukaan tietohakemisto voi sisältää sanomakuvauksia, joiden mukaisin sanomin ollaan yhteydessä tiettyyn palveluhakemistossa kuvattuun palveluun. Palveluhakemistojen tarkoitus on tehostaa palvelujen löytymistä esimerkiksi portaalien hakukoneilla etsittäessä. Suosituksessa korostetaan yhteisten sanastojen luomisen merkitystä palveluiden ja sovellusten yhteentoimivuuden rakentamisessa. Sanastot luokitellaan suosituksessa sovellusaluekohtaisiin, sovellusalue riippumattomiin sekä sanastoviitekehyksiin [38]. Valtiovarainministeriö on julkaissut myös ohjeluonnoksen yhteiskäyttöisten sanastojen nimeämiskäytännöistä [39].

Salminen [34] määrittelee webiin liittyen metatiedon tiedoksi, joka auttaa ymmärtämään, prosessoimaan ja yhdistämään webin tietoresursseja. Metatiedolla voidaan Salmisen mukaan kuvata esimerkiksi dokumentin tai sen osan sisällön rakennetta ja merkitystä, mutta myös dokumenttiin liittyviä teknisiä tietoja tai esimerkiksi henkilöresursseja. Rakenteisten dokumenttien käyttöä ja kehittämistä julkishallinnossa tutkinut Salminen jaottelee dokumenttien metadatan mm. bibliografiseen, semanttiseen, kontekstuaaliseen ja rakenteelliseen metadataan.

Semanttinen metatieto voidaan esittää Salmisen mukaan ontologioiden avulla ja rakenteellinen metatieto XML skeemalla. Kontekstuaalinen metatieto kuvaa prosesseja ja dokumentteihin liittyvää organisaatioympäristöä. Salminen toteaa, että bibliografinen metatieto voidaan esittää esimerkiksi Dublin Core -metatietostandardilla. Dublin Core (DC) on 15 ominaisuutta (nimeke, tekijä, aihe, kuvaus, julkaisija, muu tekijä, aikamääre, laji, formaatti, identifiointitunnus, lähde, kieli, suhde, kattavuus, oikeudet) sisältävä webin dokumenttien kuvaukseen kehitetty standardi.

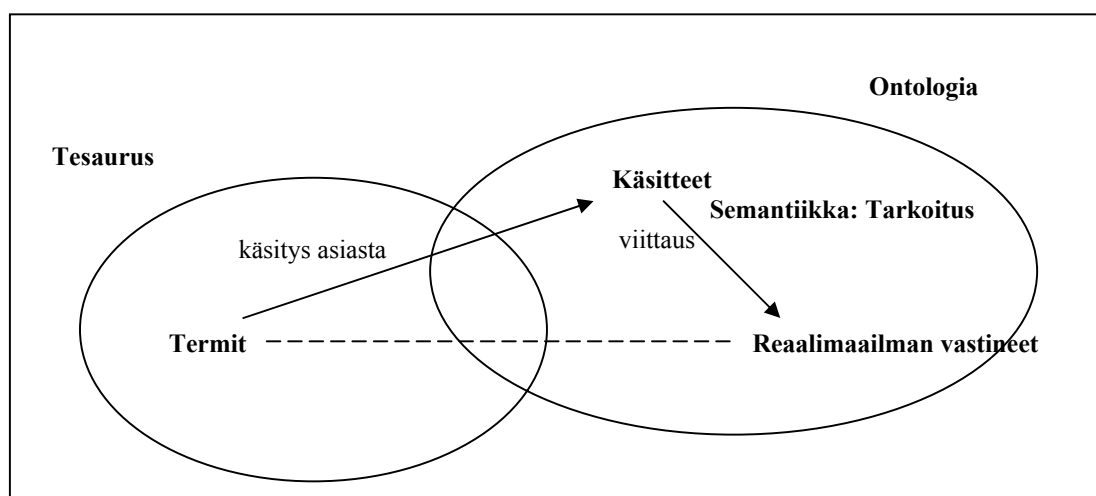
LOMv1.0, IEEE LTSC:n Learning Object Metadata -standardi, valmistui vuonna 2002, liite 3. Standardin tarkoitus on helpottaa oppijoiden, opettajien ja tietokoneohjelmien oppisisältöön kohdistuvaa etsimistä, arviointia, hankkimista ja käyttöä. Standardi on monipuolinen ja kattava kuvaus oppisisällön (oppimisaihion) ominaisuuksista, joiden muodostama käsitteellinen tietorakenne sisältää runsaat 120 tietoalkioita. Kansallisessa standardin käyttöönotossa niistä suositellaan käytettäväksi noin puolet [33]. Tarkastelen LOM:aa lähemmin luvussa 8.

Tesaurus-termi on kreikkaa ja tarkoittaa sananmukaisesti aarrearkkia. Yleisesti käytetty tesaurusta vastaava suomenkielen termi on sanasto tai asiasanasto. Sanasto on suunnit-

teltu jonkun alan tai alojen kontrolloiduksi sanastoksi tiedonhakua tai tietoavaruudessa navigointia varten [20]. YSA (Yleinen suomalainen asiasanasto) on suomenkielinen, kaikki tieteen- ja tiedonalat kattava yleissanasto, joka sisältää eri alojen yleisimmät termit. Sanastossa on myös maantieteellisiä nimiä ja luetteloita mm. kasvien, urheilulajien ja eläinten nimistä. YSA:a on käytetty tietokannoissa vuodesta 1987 lähtien [18].

Kuvassa 6 on esitetty Dacontan näkemyksen mukainen sanaston ja ontologian välinen suhde [20]. Kamut-2 projektin loppuraportissa todetaan, että sanastossa termin merkitystä ei selitetä, vaan niissä riittää tieto siitä, onko termi erotettavissa semanttisesti muista termeistä [20]. Raportissa todetaan edelleen, että sanasto keskittyy käsitteisiin (termeihin) ja niiden välisiin yksinkertaisiin semanttisiin suhteisiin ja että ontologia puolestaan esittää monimutkaista semantiikkaa ja monipuolisia suhteita vahvalla, rikkaalla semantiikalla ja kykenee esittämään myös täysin vapaasti määriteltävissä olevia, mielivaltaisia suhteita [20]. Kuvan 6 tapauksessa sanastoilla siis tarkoitetaan lähinnä termien (jotka kuitenkin samaistetaan käsitteisiin) luetteloita, joihin ei ole liitetty termien merkityksiä kuvailevaa tietoa. Luvussa 3.2 esitetty, tietojenkäsittelytieteessä käytetty käsitteen määritelmä määrittelee käsitteen kohdealuetta koskevasta tietämyksestä muodostuvaksi rakenteelliseksi kokonaisuudeksi, jonka symbolia kutsutaan termiksi. Lisäksi todetaan, että käsitteen merkitys on se kuvattavan maailman ilmiö, jota käsitteen avulla halutaan ilmaista. Tietojenkäsittelytieteen käsitteen-määritelmä kiinnittää puolestaan käsitteen sisällön. Kuvan 6 tapauksessa käsite puolestaan saa merkityksensä vasta jonkin ontologian kautta.

Kuva 6. Sanaston (tesaurus) ja ontologian suhde Dacontan mukaan [20].



Yleiseen suomalaiseen asiasanastoon sisältyy asiasanojen välisiä suhteita eli viittauksia laajempiin, suppeampiin ja rinnakkaisiin termeihin. Lisäksi siihen sisältyvät asiasanoihin liittyvät selitykset ja ryhmäkoodit, jotka osoittavat, mihin tiedon aloihin asiasana ensisijaisesti liittyy.

YSA:a ylläpidetään Helsingin yliopiston kirjastossa. Sen avulla kuvaillaan FENNICA-kansallisbibliografiatietokantaan mm. kaikki Suomessa ilmestyvä kirjallisuus sekä ARTO-tietokantaan kotimaisissa lehdissä ilmestyvää artikkeliaineistoa. YSA käsittää noin 14 000 varsinaista asiasanaa ja runsaat 3 000 ohjaustermiä, jotka ohjaavat hyväksytyn asiasanan käyttöön. YSA toimii myös eräänlaisena kattosanastona, jonka määrittelemää hierarkiaa tarkentavat eri alojen erikoissanastot, kuten kielitieteen asiasanasto [18].

WordNet on verkosta ladattava tai siellä käytettävä, Princetonin yliopiston kognitiivisen laboratorion kehittämä, englanninkielisten sanojen viittaussysteemi. Sen sisällön suunnittelu perustuu psykolingvistiseen teoriaan ihmisen kielellisestä muistista. Englanninkieliset substantiivit, verbit, adjektiivit ja adverbit on organisoitu synonyymijoukkoihin siten, että kukin joukko esittää yhtä allaolevaa käsitettä. Erilaiset suhteet linkittävät synonyymijoukkoja [37]. Sanoista on talletettu rinnastuksen lisäksi myös niiden ylä- ja alakäsitteet, osasuhteita, antonyymejä jne. WordNet voidaan integroida ontologioihin ja tietosysteemiin viittaamalla siihen verkossa. Guarino et al. ovat analysoineet OntoClean-metodilla WordNetin ylätasen luokittelussa. He toteavat, että analyysin tulos osoittaa merkittäviä puutteita WordNetin ylätasen luokittelussa ja esittävät WordNetin huolellista ontologista kehittämistä OntoClean metodin pohjalta [10].

Valtion tietotekniikan rajapintasuosituksissa XML-teknologiat ovat keskeisessä asemassa [38]. XML:n hyödyntämisen ytimeksi todettiin tarkoituksenmukaisten ja yhteiskäyttöisten sanastojen luonti ja käyttö. Sanastojen käytön tavoitteena on parantaa julkishallinnon tietotekniikan keskinäistä yhteentoimivuutta luomalla pohja tietorakenteiden merkityksen täsmälliselle dokumentoinnille, modulaarisuudelle ja uudelleenkäytettävyydelle.

XML sanastolla tarkoitetaan tietyn tyyppisessä XML-dokumentissa mahdollisesti esiintyvien elementtien ja attribuuttien nimiä. Suppeimmassa merkityksessä sanastolla yleensä tarkoitetaan vain nimiä esittäviä merkkijonoja. Laajemmassa merkityksessä sanastoon liitetään sanojen merkityksen kuvaavaa tietoa. Julkishallinnon sanastojen nimeämisohjeluonnoksessa sanastolla tarkoitetaan sanastotermeistä koostuvia dokumentoituja luetteloi-

ta, joita käytetään XML-rakenteissa käytettävien tietoelementtien luetteluun ja määrittelyyn [39]. Tarkastelen lähemmin sanaston luomista ja kuvaamista luvussa 7.

4.2 Mallintamisprosessi

Miten pitäisi rakentaa ontologia ja sitä käyttävä kohdealueen käsitteellinen malli, jotta niiden käsitteillä rakennetut sovellukset olisivat skaalautuvia ja uudelleenkäytettäviä sekä yhteistoiminnallisia muiden systeemien kanssa? Mitkä käsitteet pitäisi valita malliin ja miten ne pitäisi formuloida, jotta systeemien sisältämä tieto olisi helposti löydettävissä ja käsiteltävissä?

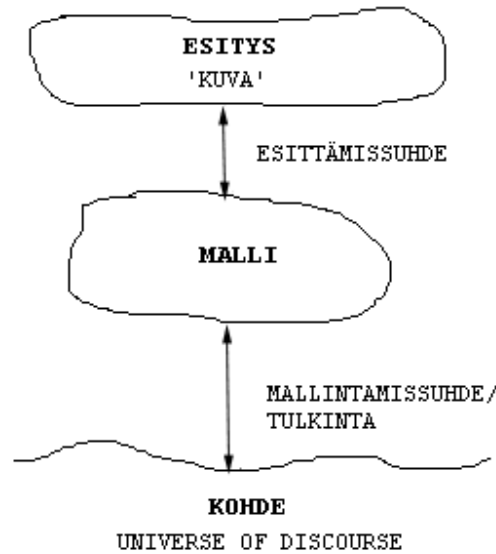
Tässä kuvattava mallintamisprosessi perustuu Kangassalon luentoihin [24] ja COMIC-projektin tuloksiin [21]. Esimerkiksi Methontology- ja Enterprise Ontology -projekteissa on noudatettu pitkälti samoja menetelmiä [27].

Mallintamisprosessi alkaa mallintamisen tarkoituksen ja tavoitteiden täsmentämisellä. Koska käytettävä tarkastelukulma ja tarkkuustaso määräytyvät mallintamisen tarkoituksen perusteella, samasta kohdealueesta eri tarkoituksiin tehdyt käsitteelliset mallit poikkeavat toisistaan.

Kuva 7 esittää mallintamisprosessia. Mallintaja tunnistaa kohdealueesta (KOHDE) ne käsitteet (kohteet ja ilmiöt) ja niiden väliset suhteet, jotka tuntuvat relevanteilta mallintamisen tarkoituksen ja tavoitteiden kannalta. Tällöin muodostuvat käsiterakenteet ja niihin pohjautuva abstrakti käsitekaavio, jota kuvassa 7 kuvataan MALLI-alueella. Mallintamissuhde osoittaa tapaa, jolla mallissa on ilmaistu kohdealueesta tunnistetut käsitteet ja niiden väliset suhteet. Tulkinta on mallintamissuhteen käänteinen suhde.

Abstraktin käsitteellisen mallin konkretisointi jollakin kuvauskielellä tuottaa tulokseksi rakenteen, jota kuvassa on merkitty termillä ESITYS. Mallin ja esityksen välinen esityssuhde määrää, millä tavalla mallin komponentit ilmaistaan esityksessä.

Kuva 7. Käsitteellisen mallintamisen prosessi [24].



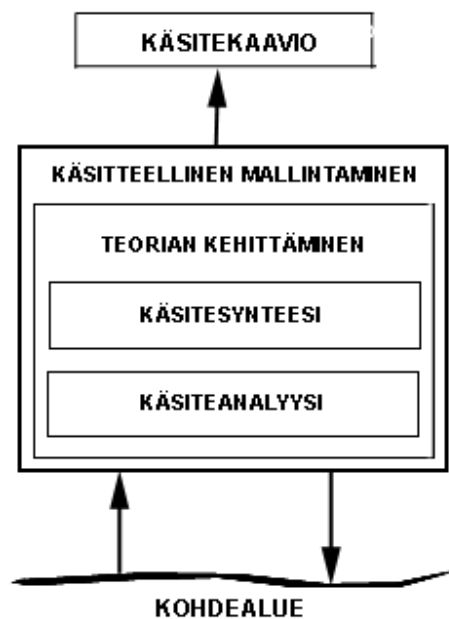
Mallintamisprosessi voidaan jakaa käsiteanalyysiin, käsitesynteesiin ja teorian kehittämiseen. Käsiteanalyysin aikana kartoitetaan kohdealueella käytössä oleva käsitteistö, analysoidaan sen ominaisuudet sekä poistetaan epätasällisyydet ja ristiriitaisuudet. Tarkoituksena on saada kohdealueen käsitteistö dokumentoiduksi ja kohde kuvatuksi tunnistettua käsitteistöä käyttäen. Synteesivaiheessa selvitetään mahdollisuudet kehittää ja parantaa käsitteistön laatua ottaen huomioon sekä mallintamisen tarkoitus että käsitteistön muodolliset ominaisuudet.

Käsitteistöä voidaan pyrkiä yksinkertaistamaan, selkeyttämään ja tekemään kokonaisuuden kannalta tarkoituksenmukaisemmaksi säilyttäen kuitenkin tarvittava käsitteellinen erottelukyky. Synteesivaiheen aikana muodostetaan usein yleistyksiä ja uusia luokituksia. Vaiheen tuloksena syntyvä käsitekaavio kuvaa alkuperäisen kohdealueen suunniteltua muutosta. Kun uusi käsitekaavio hyväksytään, oletetaan kohdealueen toimivan sen mukaisesti, ts. on sitouduttu kuvattujen käsitteiden merkityksiin ja käsitteiden välisiin suhteisiin.

Synteesivaiheen aikana voidaan käsitekaaviosta mahdollisesti tunnistaa laajoja säännönmukaisuuksia, joiden huomioonotto johtaa kohdealuetta koskevan tietämyksen niin laajaan uudelleenorganisointiin, että voidaan puhua kohdealuetta koskevan teorian kehittämisestä. Kuvassa 8 on havainnollistettu teorian kehittämisprosessia.

Käsitteellisessä mallintamisessa ovat tietojenkäsittelyn teknisestä näkökulmasta katsottuna kiinnostuksen kohteina käsitteen sisäinen rakenne, käsitteiden käyttö kohdesysteemin kuvauksessa sekä käsitteitä edustavien arvojen laskennallinen määräytyminen.

Kuva 8. Käsitteellinen mallintaminen: teorian kehittäminen.



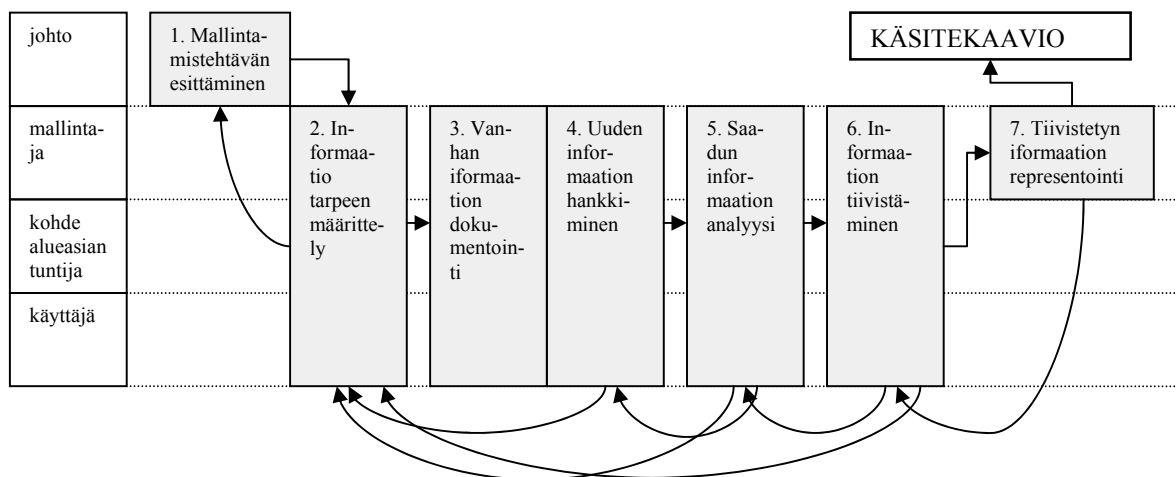
Kuvassa 9 on kuvattu tarkemmin mallinnusprosessi. Kuvassa esitetty kaavio on piirretty Kangassalon/Marjomaan esittämien mallinnusvaiheiden perusteella [24]. Kaavion tulkintaan ovat luentojen lisäksi vaikuttaneet kirjoittajan omat kokemukset mallintamistyössä.

Prosessi etenee iteroiden kunnes mallintamisen tavoite on saavutettu huomioon ottaen käytettävissä olevat resurssit. Mallintamistehtävässä rajataan (1) mallinnettava kohdealue sekä esitetään mallintamisen tavoitteet ja tarkoitus. Tavoitteena voi olla esimerkiksi toimintaprosessin kehittäminen. Tarkentamalla prosessin kuvaus riittävälle tasolle, selviävät kuvauksesta myös prosessissa käytetyt käsitteet. Tällöin alkaa hahmottua kohdealueesta mallintamistehtävän ja tarkoituksen kannalta relevantti, tarvittavan informaation määrä ja sen sisältö (2). Tarvittava tietämys täsmentyy ja saattaa lisääntyä mallinnusprosessin edetessä.

Vanhan informaation dokumentointivaiheessa (3) pyritään kartoittamaan käytössä oleva, tarkasteltavaa kohdealuetta kuvaava käsitteistö. Tällöin mallintaja pyrkii kuvaamaan kohdealueella yleisesti käytössä olevat ja sen sovelluksissa esiintyvät käsitteet. Koska eri käyttäjien ja toimijoiden käsitteet saattavat poiketa huomattavastikin toisistaan, on mallintajan syytä kuvata useamman käyttäjän näkemykset samoista käsitteistä. Kohdealueella saattaa esimerkiksi olla käytössä lukuisia määriä lomakkeita, joissa käytetyn terminologian merkitystä ei ole yhteisesti pohdittu eikä siitä ole yhteisesti päätetty. Lomakkeen laatija on kehittänyt lomakkeen omia tarpeitaan vastaavaksi kuvitellen lomakkeen täyttäjän ja käsittelijän käsitteellistävän asiat ja ilmiöt itsensä lailla. Samoin kohdealueen toimintoja hallinnoivissa sovelluksissa saattaa olla toisistaan poikkeavia määrittelyjä ja termejä samaa kohdetta tai ilmiötä kuvaavista käsitteistä.

Olemassa olevan käsitteistön kartoitus ja analysointi paljastavat usein aukkoja ja epäselvyyksiä, jotka vaativat uuden informaation hankkimista (4) mallintamisen tarkoituksen saavuttamiseksi. Informaation tiivistäminen (6) (käsitesynteesi) sekä tiivistetyn informaation esittäminen (7) saattavat aiheuttaa informaatiotarpeen määrittelyn täsmentämistä yhä uudelleen ja uudelleen. Tiivistetyn informaation representointi tarkoittaa tässä käsite- rakenteiden kuvaamista.

Kuva 9. Mallintamisen vaiheet.



Kun kohdealueyhteisössä on saavutettu yhteisymmärrys käsitteiden sisällöstä ja kattavuudesta suhteessa prosessin tavoitetilään tai kun mallintamiseen varatut resurssit on käytetty, päätetään mallintamisprosessi ja muodostetaan kohdealueesta siihen mennessä hanki-

tulla tietämyksellä sitä kuvaava käsitekaavio (kuva 1). Koska ympäristön ja tarpeiden muuttuessa syntyy uusia käsitteitä ja jotkin käytössä olevat käsitteet saattavat muuttua tai poistua, on käsitekaaviota päivitettävä. Siksi sen huolellinen dokumentointi ja talletus on tärkeää.

4.3 Yhteenveto

Tässä tutkielmassa käsiteltyjen esimerkkien valossa tutkijoiden parissa ollaan pitkälti yksimielisiä ontologian merkityksestä tiedon löydettävyyden ja uudelleenkäytön alueella sekä yleisesti tietämyksenhallinnan alueella; ontologia on formaali, eksplisiittinen, käsitteellistämisen spesifikaatio [7, 11, 30]. Jonkin kohdealueen ontologian katsotaan myös samaistuvan ko. kohdealueen sisältöteoriaan [11, 5]. Mikäli sisältöteorialla tarkoitetaan kohdealueen käsitteellistä mallia, tutkijoiden joukko laajenee [16, 24].

Guarino et al. ja Kangassalo painottavat yleisesti käsite-termiä, kun taas muut puhuvat useimmiten sanastosta [11, 30]. Myös Guarino käyttää sanasto-termiä määritellessään tietämyksenhallintaan liittyen ontologian loogiseksi teoriaksi, joka pitää kirjaa formaalin sanaston tarkoitetuista merkityksistä [14].

Guarino et al. toteavat [14], että kielellä L esitetty ontologia O approksimoi käsitteellistämistä C , jos on olemassa sellainen ontologinen sitoutuminen $K = \langle C, I \rangle$, että kielellä L esitettävät mallit ovat sopuoinnussa ontologian O malleihin sisältyvään ontologisen sitoutumisen K kanssa. Gruberin mukaan [11] ontologinen sitoutuminen tarkoittaa käytännössä sopimusta käyttää jotakin sanastoa tavalla, joka on ehyt suhteessa jonkin ontologian spesifioimaan teoriaan. Mikäli Gruber viittaa edellisessä lauseessa ontologialla ja sanastolla toisiaan vastaavaan käsiteavaruuteen, voitaneen Guarino et al.:n ja Gruberin näkemyksiä pitää yhtenevinä.

Formaali, eksplisiittinen sanasto on esitetty aina jollakin keinotekoisella kielellä, jota kone pystyy ymmärtämään. Wrobel korostaa, että sanasto ei tarkoita syntaktisten termien joukkoa vaan käsitteiden systeemiä [46]. Tarkoituksenmukaisella sanastolla voidaan olettaa tarkoitettavan sanastoa, jonka sisältämien termien merkitys on yleisesti hyväksytty, ts. sen merkityksiin on sitouduttu.

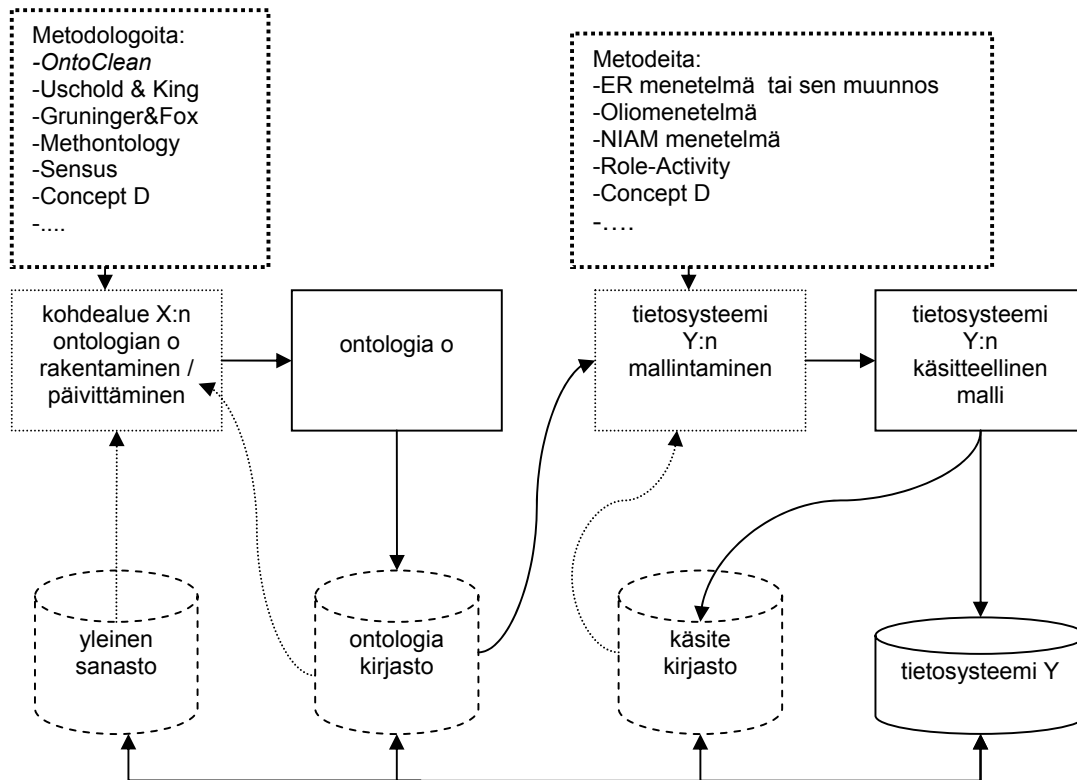
Kohdealueesta muodostettu ontologia ja siihen pohjautuva sovelluksen käsitteellinen malli on riippuvainen siitä, miten olemme käsitteellistäneet kohdealueen oliot ja ilmiöt sekä niiden väliset suhteet. Kuva 10 pyrkii kuvaamaan ontologioiden ja käsitteellisen mallinta-

misen roolien suhteita tietosysteemien elinkaaren aikana. Mikäli tietosysteemin suunnittelun aloitusvaiheessa ei ole käytettävissä mitään valmista ontologiaa, joudutaan relevantit käsitteet ja niiden suhteet mallintamaan, jolloin syntyy ainakin jokin osa kohdealueen ontologiasta. Yleisellä sanastolla tarkoitetaan kuvassa esimekiksi WordNetiä tai YSA:a.

Oikein valitut ontologioiden rakentamismetodit sekä yleisesti hyväksytyjen sanastojen käyttö mahdollistavat ontologioiden uudelleenkäytön tietosysteemien elinkaaren eri vaiheissa ja uusien systeemien rakentamisessa [13]. Jotakin tarkoitusta varten kuvattu tietosysteemin käsitteellinen malli kuvataan mallia vastaavan kohdealueen kohdealueontologiassa määritellyllä terminologialla.

Tiedon käsitteellinen rakenne on usein luonteeltaan pysyvämpää kuin teknologia, jota käytetään manipuloimaan dataa [21]. Kangassalon mukaan oikean tiedon määrittely ja ylläpito ovat pääkysymyksiä suunniteltaessa tietosysteemejä ja niiden uudelleenkäyttöä. Käsitekaavio yhdistää tietojärjestelmiä. Käsitekaavio määrittelee sen, miten asiat liittyvät toisiinsa. Sen jostakin osasta tehdään mahdollisesti tietokanta, osa käsitekaaviosta sisältyy kohdesysteemiä kuvaaviin ohjelmiin ja hakemistoihin.

Kuva 10. Ontologian ja tietosysteemin suhde.



Yhteisesti käytettävät sanastot perustuvat yhteiseen käsitteellistämiseen. Käsite- rakenteet ovat sanastojen muodostamisen perusta. Kohdealueen ominaisuuksien luokittelu luokittelee siinä käytetyt käsitteet. Käsitteellinen mallintaminen kattaa sekä käsite- rakenteiden, ontologioiden, prosessien käsitteellisten mallien että niiden hallintaan määriteltyjen tietosysteemien käsitekaavioiden rakentamisen. Metatietoina voidaan pitää kaikkia edellä mainittuja kuvauksia kohdealueesta.

5. Concept-D

Käsitekaavioiden laatimiseen on kehitetty useita graafisia kuvauskieliä. Tunnetuimpana voidaan pitää ER (Entity Relationship) -mallin kuvaamiseen kehitettyä notaatiota ja sen muunnelmia, mm. UML (Unified Modelling Language)-notaatiota. Muita kuvauskieliä ovat mm. OR (Object Role Modelling) [2]-mallia kuvaava kieli sekä käsitteen intension kuvaamiseen kehitetty Concept-D-kieli.

ER-malli perustuu sovellusalueen olioiden ja ilmiöiden esiintymien kuvaamiseen kohteina, attribuutteina ja suhteina. Kohteet, attribuutit ja suhteet muodostavat siis ER-mallin ontologian. ORM:n ontologia rakentuu objekteista, objektityypeistä, objektien välisistä binäärirelaatioista sekä objektien välisistä suhdetyypeistä. ER-mallin pohjalle perustuva UML kehitettiin olioparadigman mukaisen ohjelmiston kuvaamiseen. UML:een perustuvilla kaavioilla kuvataan suositusten [39] mukaisesti myös julkishallinnon sanastot. Sekä ER- että UML-mallit perustuvat joukko-oppiin. ER-, OR- ja UML-mallien tarkastelua ei ole mahdollista sisällyttää tähän esitykseen. Concept-D -kielen esittelen tässä luvussa.

Concept-D-kieli [22] on Kangassalon kehittämä, intensionaaliseen lähestymistapaan perustuva graafinen kuvauskieli käsite rakenteiden esittämiseen. Kielellä kuvataan intensionaalisen sisältymissuhteen avulla käsitteisiin sisältyvän, kohdealuetta koskevan tietämyksen koostumista muista käsitteistä ja tietämysalkioista [22]. Tapio Niemi on kehittänyt Cosmos-editorin Concept-D-kielellä kuvattavien käsite rakenteiden editointiin ja kuvattujen käsitteiden pohjalta kehitettävän tietokantakaavan muodostamiseen [30]. Tässä luvussa esittelen Kangassalon kehittämän kielen perusteet lyhyesti. Kuvatut käsite rakenteet on tehty Cosmos-editorilla. Sekä peruskäsite että johdettu käsite kuvataan teknisistä syistä tässä tutkielmassa yhdellä alleviivauksella.

5.1 Peruskäsite ja sen kuvaaminen

Peruskäsite on käsite, jota ei annetussa tai muodostettavassa käsitejärjestelmässä voida kuvata käsitejärjestelmän muiden käsitteiden avulla, eli peruskäsite ei sisällä muita käsitejärjestelmän käsitteitä.

Peruskäsitteet kuvataan luonnollisella kielellä käyttämättä muodostettavan käsitteistön käsitteitä ilmaisevia termejä. Kielen syntaksin mukaan peruskäsite kuvataan graafisesti ilmoittamalla sen nimi kaksoisviivalla alleviivattuna ja liittämällä nimeen tarvittavat lisä-

määreet. Samassa käsitejärjestelmässä käsitteiden nimien tulee poiketa toisistaan. Peruskäsitteeseen liittyvä määrelista sisältää viittauksen sen sanalliseen kuvaukseen (D_i), käsitettä edustavien arvojen joukon kuvaukseen (V_i), käsitteen esiintymää vastaavaan havaintomalliin (Q_i) sekä käsitteeseen mahdollisesti liittyvään operationalisointiproseduriin tai sen kuvaukseen (P_i). Peruskäsitteeseen voidaan liittää aikamääreitä, esiintymisehtoja ja erilaisia rajoituksia. Tekstissä käsitteen arvojoukkoon viitataan merkinnällä $V_{\langle \text{käsitteen nimi} \rangle}$ ja yhteen arvoon merkinnällä $v_{\langle \text{käsitteen nimi} \rangle}$.

Kaikkien mahdollisten käsitettä edustavien arvojen joukkoa sanotaan käsitteen arvojoukoksi. Joissakin tilanteissa erotetaan toisistaan potentiaalisten arvojen joukko eli kaikkien mahdollisten arvojen joukko ja aktuaalisten arvojen joukko eli tietyllä hetkellä olemassa olevien arvojen joukko.

Peruskäsitettä edustavien arvojen kuvaus V

Arvolla tarkoitetaan käsitettä vastaavaa havainnon, mittauksen, luokituksen, päättelyn tai laskutoimituksen symbolein esitettävää tulosta. Arvojoukko kuvataan arvojen, arvonnäilyfunktion ja arvopoistofunktion avulla, taulukko 1.

Taulukko 1. Peruskäsitteen arvojoukon kuvaaminen.

Arvot	arvojen tyyppi
	arvojoukon rajat
	arvojoukon koko: arvojen aktuaalisten esiintymien lukumäärä tietyllä ajanhetkellä tai määrättyllä aikavälillä
	historia: aktuaalisen arvojoukon käyttäytyminen ajan kuluessa, arvojen säilyttäminen jne.
Arvonnäilyfunktio	funktion toimintaehto eli 'heräte'
	funktion määrittely, joka määrittelee arvoja tuottavan mekanismin riittävällä tarkkuudella
	funktion toimintafrekvenssi ja ajanjakso
	funktion muutosehto, kuka muutoksen saa tehdä tai kenen lupa muutoksen tekemiseen tarvitaan
Arvopoistofunktio	funktion toimintaehto
	valintasääntö: mikä tai mitkä arvot kulloinkin poistetaan, mikäli valintasääntö ei riipu poistotilanteesta eikä ole poiston suorittajan määrättävissä
	funktion toimintafrekvenssi ja ajanjakso
	funktion muutosehto; tässä voidaan myös ilmoittaa se, kuka muutoksen saa tehdä tai kenen lupa muutoksen tekemiseen tarvitaan

5.2 Johdettu käsite ja sen kuvaaminen

Johdettu käsite on käsite, joka on muodostettu määritelmän osoittamalla tavalla muista saman käsitejärjestelmän käsitteistä. Käsitteen määritelmä kuvaa sen, millä tavalla määriteltävään käsitteeseen sisältyvä tietämys muodostetaan määrittelevistä käsitteistä ja määritelmään itseensä sisältyvistä tietämysalkioista. Johdettu käsite kuvataan ilmoittamalla käsitteen nimi alleviivattuna yhdellä viivalla, johon liittyy viittaus kyseisen käsitteen sanalliseen

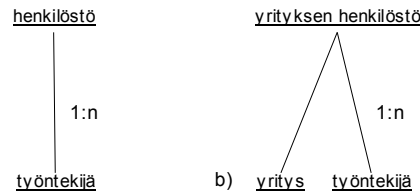
kuvaukseen (D_i), mahdolliseen havaintomalliin (O_i) ja käsitettä edustavien arvojen joukon kuvaukseen (V_i). Johdetun käsitteen määrittelytapa käy kuvauksessa ilmi tavasta, jolla määrittelevät käsitteet liitetään määriteltävän käsitteen kuvaukseen. Johdettu käsite voidaan määrittellä aggregoinnilla, yleistyksellä tai transformoinnilla.

Aggregointi eli koostaminen

Aggregoinnissa määriteltävä käsite muodostetaan liittämällä määriteltävän käsitteen nimeen käsitteen tunnusmerkeiksi intensionaalisen sisältymissuhteen avulla yksi tai useampia määritteleviä käsitteitä määreineen. Sisältymissuhteeseen liitetään määrittelevää käsitettä koskevat ehdot, rajoitukset tai ehdolliset rajoitukset.

Kuvassa 11 a) on esimerkki tapauksesta, jossa henkilöstö-käsite on kuvattu käyttäen vain sisältymissuhdetta, yhtä määrittelevää käsitettä työntekijä sekä lukumäärärajoitusta. Suhde $1 : n$ ilmaisee, että henkilöstöön voi kuulua luvun n ilmoittama määrä työntekijöitä. Kuvan 11 b) kohdan esimerkissä 'yrityksen henkilöstö'-käsitteeseen sisältyvät intensionaalisesti sekä yritys-käsite että työntekijä-käsite. Kuvan 11 käsite rakenteet voisivat sisältyä esimerkiksi kohdealueontologiaan, jossa kohdealueena on yritys tai sen henkilöstöhallinto.

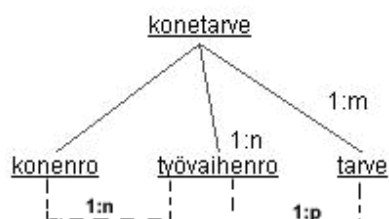
Kuva 11. Aggregoimalla muodostette johdettu käsite.



Kuvan 12 esittämä käsite rakenne on muodostettu useammasta määriteltävään käsitteeseen intensionaalisesti sisältyvästä käsitteestä, joiden välillä vallitsee ehtoja. Esimerkkikuvassa konetarve-käsitteen määrittely koskee tilannetta, jossa yhtä konetta käytetään monessa eri työvaiheessa ja yhdessä työvaiheessa käytetään vain yhtä konetta. Mikäli yhtä konetta käytetään monessa eri työvaiheessa ja kussakin työvaiheessa käytetään useaa konetta, pitää määritelmää täydentää. Siihen pitää lisätä rajoitus, jonka mukaan tarve-lukuja voi liittyä vain niihin työvaihenumeroihin, jotka liittyvät ainoastaan yhteen kone-numeroon. Kuvan 12 esimerkissä konenumero yksilöi käsitteen esiintymän. Kaikki tarve-määrittelyt käsitteen esiintymässä koskevat samaa konetta, joten työvaihenumero erottelee eri tarve-

lukumäärät toisistaan. Konetarve-esimerkki voisi olla esimerkki käsitteestä, joka on määritelty yrityksen sovellusalueontologiassa. Määriteltävälle käsitteelle voidaan lisäksi määrittellä yksi tai useampia, määrittelevistä käsitteistä muodostettuja tunnisteita ja niihin liittyviä kantamarajoja.

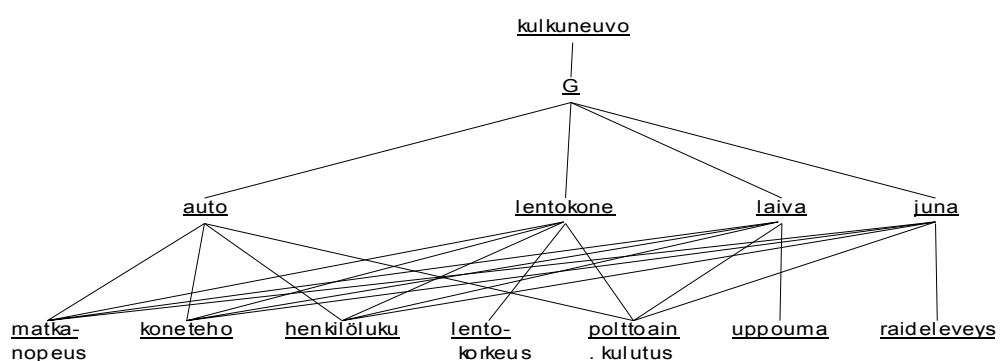
Kuva 12. Johdetun käsitteen määrittely aggregoimalla useasta käsitteestä.



Yleistys

Käsite määritellään yleistystä käyttäen liittämällä määriteltävän käsitteen nimeen käsitteen tunnusmerkeiksi joko kaikki tai vain valittu joukko määrittelevien käsitteiden yhteisiä tunnusmerkkejä. Yleistys voi olla tyypiltään rajoittamaton, eksplisiittinen tai implisiittinen. Rajoittamattomassa yleistyksessä määriteltävän käsitteen tunnusmerkeiksi otetaan kaikki määrittelevien käsitteiden yhteiset tunnusmerkit.

Kuva 13. Kulkuneuvo-käsitteen evaluointi .



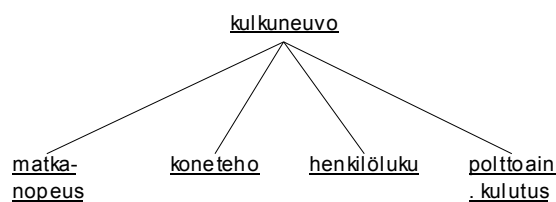
Eksplisiittisessä yleistyksessä rajoituslauseke ilmaisee, mitkä määrittelevien käsitteiden yhteiset tunnusmerkit ainakin otetaan määriteltävän käsitteen ominaisuuksiksi. Impli-

siittäisessä yleistyksessä rajoituslauseke ilmaisee, mitkä määrittelevien käsitteiden yhteisistä tunnusmerkeistä jätetään pois määriteltävän käsitteen tunnusmerkkien joukosta. Määrittelevinä käsitteinä voi olla yleistystä käytettäessä yksi tai useampia käsitteitä.

Kuvassa 13 määritellään kulkuneuvo-käsite erityyppisten kulkuneuvojen rajoittamattomana yleistysenä. Käsitteen määrittelee määrittelevien käsitteiden intensiojoukkojen leikkausjoukkoon kuuluvat kulkuneuvojen ominaisuudet. Evaluoinnin tulos on esitetty kuvassa 14.

Yleistyksen syvyydellä tarkoitetaan niiden määrittelytasojen lukumäärää, jotka yleistystä muodostettaessa otetaan huomioon määrittelevistä käsitteistä. Jos yleistysten syvyys on pienempi kuin määrittelyhierarkian tasojen lukumäärä, niin määrittelevien käsitteiden sisältö otetaan huomioon vain yleistysten syvyyden ilmoittamalle tasolle asti. Tällöin kokonaisuuden kannalta merkityksettömät yksityiskohdat voidaan häivyttää pois. Kulkuneuvo-käsitteen määrittelyssä määrittelevät käsitteet määriteltiin huomioimalla vain yksi määrittelytaso, kuva 14.

Kuva 14. Kulkuneuvo -käsite



Transformointi eli arvon muunnos

Transformoinnissa käsite määritellään kuvaamalla määriteltävää käsitettä edustavan arvon määräämistapa tai laskukaava, jonka mukaisesti arvo johdetaan tunnusmerkeinä olevia määritteleviä käsitteitä edustavista arvoista. Määritelmässä kuvataan myös määriteltävän käsitteen ja määrittelevien käsitteiden väliset, samoin kuin määrittelevien käsitteiden keskinäiset ehdot, rajoitukset ja ehdolliset rajoitukset.

5.3 Ehdot ja rajoitukset

Ehdot, rajoitukset ja ehdolliset rajoitukset ovat käsiterakenteen osia. Niillä voidaan määritellä käsitteen esiintymän olemassaolo ehtojen ollessa tosia tai erikoistapaus rajoitus-

ten ollessa tosia, tai sulkea esiintymän olemassaolo pois ehdollisten rajoitusten ollessa tosia. Niiden tunnusmerkkien joukkoa, johon ehto viittaa, kutsutaan viittausyhteydeksi (reference context). Viittausyhteyden perusteella on voitava testata ehdon totuusarvo. Ehto ja rajoitus sekä ehdollinen rajoitus voivat esiintyä kukin erikseen tai kaikki yhdessä. Ehtoja ja rajoituksia koskevat tapaukset on kuvattu taulukossa 2.

Taulukko 2. Tapaukset, joita ehdot ja rajoitukset voivat koskea

Ehdot ja rajoitukset voivat koskea	
1.	Käsitettä x edustavaa arvoa
2.	Käsitteen x ilmentymien identtisyttä tai erilaisuutta
3.	Käsitteen x ilmentymään y liittyvää aikaa
4.	Käsitteen x ilmentymien lukumäärää
5.	Käsitteiden x ja y välisiä intensionaalisia suhteita
6.	Käsitteiden x ja y ilmentymien välisten suhteiden esiintymien lukumääräsuhteita
7.	Käsitteiden x ja y ekstensioiden välisiä joukko-opillisia suhteita
7.	Käsitteen x ilmentymän esiintymistä
	Lisäksi rajoitus voi koskea
9.	Ehtoja
10.	Rajoituksia

5.4 Tunnisteet

Tunniste eli tunnus K on käsitteeseen A intensionaalisesti sisältyvä käsite k tai käsitteiden k_1, k_2, \dots, k_n yhdistelmä, jolla on seuraavat ominaisuudet:

- käsitteen A ilmentymien ja tunnisteeseen K ilmentymien välillä on täydellinen 1:1 suhde,
- kaikki tunnisteeseen ilmentymät ovat erilaisia,
- tunnisteeseen ilmentymällä on aina arvo,
- käsitteiden k_1, k_2, \dots, k_n yhdistelmä on minimiyhdistelmä, joka yksilöi A :n,
- tunnituksen K vaikutusalue eli kantama (scope) on koko se käsiterakenne, joka määrittelee käsitteen A .

Tunnisteeseen vaikutusalue voidaan rajata liittämällä rajoitusmerkki poisrajattavan käsitteen yläpuolella olevaan, sisältymisrelaatiota kuvaavaan viivaan. Jos käsiterakenteessa esiintyy johdettuja käsitteitä määrittelevinä käsitteinä, niin myös määrittelevissä käsitteissä tarvitaan tunnisteita niiden ilmentymien yksilöimiseksi. Tällöin tavallisesti määrittelyhierarkiassa alempana oleva tunniste ja sen vaikutusalue sisältyvät hierarkiassa ylempänä olevan tunnisteeseen vaikutusalueeseen. Tunnisteet, jotka määrittelevät sisäkkäiset vaikutusalueet, muodostavat tunnistehierarkian.

5.6 Yhteenveto

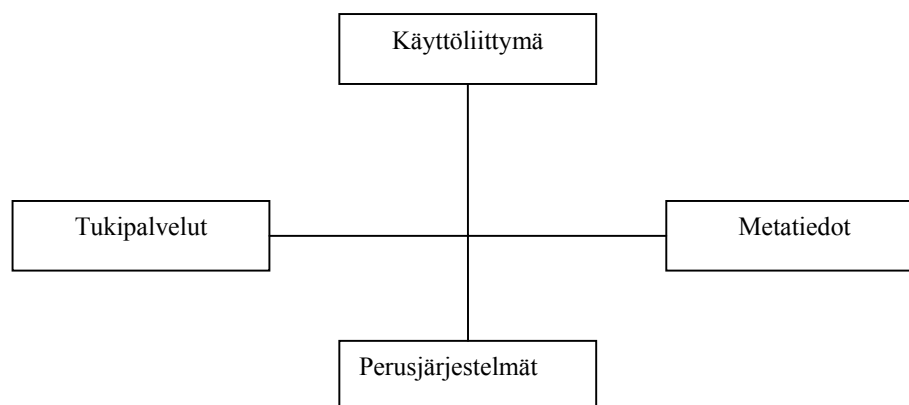
Käsitteellisessä mallintamisessa käsitteen merkitys esitetään käsiterakenteena, jossa on monenlaisia sisäisiä suhteita. Concept-D-kieli on kehitetty käsiterakenteiden graafiseen kuvaukseen. Käsitteen graafinen kuvaus helpottaa ihmistä tajuamaan käsitteen määritelmän s.o. käsitteen sisäisen rakenteen. Kuten kuva 9 osoittaa, mallintamisen tulos saavutetaan palaamalla tarvittaessa yhä uudelleen edellisiin vaiheisiin. Mallintamisen välituloksia tarkastellaan käyttäjien ja sovellusalueasiantuntijoiden kesken. Tällöin on tärkeää, että tulevat käyttäjät pystyvät hahmottamaan käsitteet, joilla järjestelmät tullaan toteuttamaan. Concept-D kielellä tehdyt kuvaukset ovat loogisia, selkeästi luettavia ja ymmärrettäviä. Pienessä tilassa pystytään ilmaisemaan monimukaisia kokonaisuuksia, joita täydennetään luonnollisen kielen sanallisilla kuvauksilla. Näyttäisi siltä, että Concept-D -kielellä esitetyt, mallintamisen tuloksena syntyneet käsiterakenteet soveltuvat XML - dokumenttityyppien malliksi.

6. XML

World Wide Web konsortio (W3C) julkisti eXtensible Markup Language (XML) 1.0 spesifikaation vuonna 1998. XML:n kehittämisen tarkoituksena oli luoda laitteisto- ja ohjelmistoriippumaton, avoin, yksinkertainen, ohjelmallisesti käsiteltävä rakenteisen tiedon kuvauskieli. Merkittävimmät ohjelmistoalan yritykset sekä monien maiden julkisten sektorien edustajat ovat olleet mukana XML-tekniikoihin pohjautuvassa erikoissanastojen ja sovellusten kehitystyössä. Tämän seurauksena XML-muotoon rakenteistettujen aineistojen käsittelylle on olemassa laaja tuki. Varsinaisia W3C:n suosituksia kutsutaan epävirallisesti XML-perheeksi, yksityisen sektorin laajan hyväksynnän saaneita määrytyksiä sanotaan XML:n de facto -standardeiksi.

XML:n ytimeksi kutsutaan standardien XML 1.0, Namespaces in XML 1.1 (nimiavaruudet) ja XML Schema (kaava, skeema) muodostamaa kolmikkoa. XML 1.0 –spesifikaatio määrittelee XML-merkkaukielen syntaksin, Namespaces in XML 1.1 määrittelee tavan, jolla XML-dokumentin elementtien nimet voidaan liittää asiayhteyteensä, joka on kuvattu XML Schema-määrytyksessä kuvatulla tavalla.

Kuva 15. Looginen tavoitearkkitehtuuri [38]



Valtiovarainministeriön määrittelemässä julkishallinnon XML-strategiassa määritellään XML:n soveltamiskohteiksi perusjärjestelmien asiointiliittymät, perusrekistereiden rajapinnat, järjestelmien välinen tiedonvaihto, tekstitiedon rakenteistaminen, monikanava-julkaiseminen, tiedon haku ja tiedon pitkäaikaissäilytys [40]. Kuvassa 15 on esitetty valtion tietotekniikan rajapintasuosituksen [38] mukaisen, sähköisessä asiointissa käytettävän loo-

gisen viitearkkitehtuurin perusosiot, joissa kaikissa XML:n ytimen perustalle rakennetut standardit näyttävät ratkaisevaa osaa.

XML on metakieli, jolla voidaan määritellä sovelluskohtaisia ja/tai kohdealuetta kuvaavia rakenteellisia, formaaleja merkkaukieliä. Kukin kieli muodostaa oman nimiavaruutensa [41], johon voidaan viitata toisesta merkkaukielestä. Muodostetuilla kielillä voidaan luoda sovellusympäristöriippumattomia standardeja määrityksiä (xml-sanastoja) jonkin kohdealueen käsitteiden, tietomallien ja asiakirjojen luomiseen, hallitsemiseen ja siirtämiseen. Mikä tahansa organisaatio tai organisaatiot yhdessä voivat luoda omia merkkaukieliään.

6.1 XML-dokumentti

XML-tiedostoa sanotaan XML-dokumentiksi, jos se voidaan jäsentää XML 1.0 – spesifikaation osoittamalla tavalla, ts. se on hyvin muodostettu (well-formed). Käytännössä dokumentti on hyvin muodostettu, kun merkkkaus on virheetöntä, dokumentissa on vähintään yksi elementti, dokumentissa on tasan yksi juurielementti sekä elementtien alku- ja loppumerkit (eng. tag) noudattavat ajatellun hierarkkisen elementtirakenteen mukaista sisäkkäisyyttä.

XML-dokumentti sisältää esittelyosan (prolog) sekä esiintymäosan (instance). Esittelyosa voi sisältää XML-määrityksen (XML declaration), dokumentin tyyppimäärityksen (document type definition) sekä kommentteja ja prosessointiohjeita. Dokumentin esiintymäosan on sisällettävä ainakin juurielementin (document root), jonka sisälle kaikki muu merkkkaus sisältyy. Sen lisäksi esiintymäosa voi sisältää kommentteja ja prosessointiohjeita. XML-dokumentti voi koostua useista entiteeteistä (entity), joista jokaisella on nimi ja sisältö. Jokainen dokumentti sisältää ainakin dokumenttientiteetin, jonka sisältönä on ko. XML-dokumentin teksti. Entiteetit luokitellaan tekstientiteetteihin (parsed entity) ja dataentiteetteihin (unparsed entity). Tekstientiteettien sisältönä on XML-tekstiä, jota dokumenttia käsittelevä ohjelma, XML-proessori (parser), tulkitsee. Dataentiteetin sisältö voi olla mitä vain, esimerkiksi dokumenttiin entiteettiviittauksen avulla liitettävä kuva. Entiteetti voi sijaita dokumentin esittelyosassa tai se voi olla talletettu omaan tiedostoonsa. Entiteettien käyttö mahdollistaa dokumentin hajautetun sijainnin ja käsittelyn verkossa. Tällöin esimerkiksi laajojen dokumenttien tuotanto voidaan hajauttaa organisaation eri yksiköihin tarkoituksenmukaisella tavalla.

Kuva 16 esittää XML-dokumenttia, jonka tietosisältönä on erään yrityksen henkilöstöä kuvaavat tiedot. Dokumentin esittelyosa sisältää julistuksen ja prosessointiohjeen. Julistus

```
<?xml version="1.0" encoding="UTF-8"?>
```

kertoo dokumenttia käsittelevälle ohjelmalle XML-version ja koodaustavan. Prosessointiohje

```
<?xml - stylesheet type="text/css" href="hallinto.css">
```

kertoo tämän dokumentin tulostuksessa käytettävän tyylitiedoston tyyppin ja nimen.

Kuva 16. henkilöstö.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml - stylesheet type="text/css" href="hallinto.css">
<henkilöstö xmlns="http://www.firma.fi/XMLSchema/henkilosto" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.firma.fi/XMLSchema/henkilosto">
  <työntekijä>
    <palvelussuhde>
      <tyyppi>toistaiseksi</tyyppi>
      <alkpvm>2001-01-01</alkpvm>
      <kk-palkka>2500</kk-palkka>
      <rooli>kouluttaja</rooli>
      <työsopimusID>0002</työsopimusID>
    </palvelussuhde>
    <henkilö>
      <hetu>111171-010k</hetu>
      <etunimi>Kaino</etunimi>
      <sukunimi>Kouluttaja</sukunimi>
      <puhelin>040-123456</puhelin>
      <email>kaino.kouluttaja@firma.fi</email>
      <kotiosoite>
        <katuosoite>Koulutie 10</katuosoite>
        <postinumero>010101</postinumero>
        <postitoimipaikka>Koulukaupunki</postitoimipaikka>
      </kotiosoite>
    </henkilö>
  </työntekijä>
  <työntekijä>
    <palvelussuhde>
      <tyyppi>toistaiseksi</tyyppi>
      <alkpvm>2000-01-01</alkpvm>
      <kk-palkka>3000</kk-palkka>
      <rooli>pomo</rooli>
      <työsopimusID>0001</työsopimusID>
    </palvelussuhde>
    <henkilö>
      <hetu>111161-010p</hetu>
      <etunimi>Pekka</etunimi>
      <sukunimi>Pomo</sukunimi>
      <puhelin>040-1234567</puhelin>
      <email>pekka.pomo@firma.fi</email>
      <kotiosoite>
        <katuosoite>Koulutie 11</katuosoite>
        <postinumero>010101</postinumero>
        <postitoimipaikka>Koulukaupunki</postitoimipaikka>
      </kotiosoite>
    </henkilö>
  </työntekijä>
</henkilöstö>
```

Kuvan 16 XML-dokumentin esiintymäosa sisältää juurielementin nimeltään henkilöstö. Juurielementillä on kolme attribuuttia: `xmlns`, `xmlns:xci` ja `xsi:schemaLocation`. Attribuutin `xmlns` arvo kertoo nimiavaruuden, jossa on määritelty dokumentin elementtien nimet. `xmlns:xci` -attribuutin arvo kertoo, että tämän dokumentin elementtien tyypit ja rakenne on kuvattu dokumentissa, jonka sijainti on ilmoitettu XML Schema instance -nimiavaruudessa määritellyn attribuutin `xsi:schemaLocation` arvona. Juurielementillä on lapsielementti `työntekijä`, joka esiintyy dokumentissa kaksi kertaa. `työntekijä`-elementillä on kaksi lapsielementtiä: `palvelussuhde` ja `henkilö`. Näillä on edelleen omia lapsielementtejään. Kaikki elementit alkavat alkumerkkauksella ja päättyvät loppumerkkaukseen, esimerkiksi `<henkilö>` ja `</henkilö/>`.

XML-dokumentti on validi, jos se noudattaa annettua dokumentin tyyppimäärittystä ts. on hyvin muodostettu ja lisäksi noudattaa sanaston ja dokumentin rakenteen osalta annettua XML DTD:tä tai XML-skeemaa. Dokumentin tyyppi voidaan kuvata XML-skeemalla (Schema) [43] tai DTD-kuvauksella [44]. DTD (Document Type Definition) -tekniikkaa on käytetty laajasti rakenteisten asiakirjojen kuvaamiseen julkishallinnossa. Tekniikkaa voidaan kuitenkin pitää väistyvänä, sen korvaa ja on jo pitkälti korvannut standardin aseman saavuttanut, DTD-tekniikkaa ilmaisuvoimaisempi XML Schema. XML Schema tukee mm. nimettyjä tietotyyppejä, abstrakteja tietotyyppejä ja periytyvyyttä. Spesifikaatio määrittelee kattavan joukon yksinkertaisia perustietotyyppejä, joista periyttämällä voidaan edelleen johtaa sovelluksen vaatima, tarkasti rajattu tietotyyppi. Myös rakenteisia tietotyyppejä voidaan määritellä ja nimetä. Konkreettiset, eri sovelluksissa käytettävät elementit voidaan sitten tyypittää kaavassa (schema) määriteltyjä tyyppinimiä hyväksi käyttäen.

Kuvassa 17 on esitetty kuvassa 16 esitetyn XML-dokumentin tyyppimäärittys. XML-dokumentin `henkilöstö.xml` tyyppiksi määriteltiin juurielementin `henkilöstö` attribuutilla `xsi:schemaLocation` dokumentissa `henkilöstö.xsd` määritelty tyyppi. Dokumentin `henkilöstö.xsd` malli on esitetty puurakenteena kuvassa 18.

Kuva 17. `henkilosto.xsd`

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.firma.fi/XMLSchema/henkilosto" elementFormDefault="qualified" attributeFormDefault="unqualified" version="0.1" xmlns="http://www.firma.fi/XMLSchema/henkilosto"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="henkilöstö">
  <xs:annotation>
```

```

    <xs:documentation>organisaation henkilökuntaan kuuluvat työntekijät</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="työntekijä" maxOccurs="5000">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="palvelussuhde">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="tyyppi" type="xs:string"/>
                  <xs:element name="alkpvm" type="xs:date"/>
                  <xs:element name="kk-palkka" type="xs:integer"/>
                  <xs:element name="rooli" type="xs:string" maxOccurs="5"/>
                  <xs:element name="työsopimusID" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="henkilö">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="hetu" type="xs:string"/>
                  <xs:element name="etunimi" type="xs:string" maxOccurs="3"/>
                  <xs:element name="sukunimi" type="xs:string"/>
                  <xs:element name="puhelin" type="xs:string"/>
                  <xs:element name="email" type="xs:string"/>
                  <xs:element name="kotiosoite">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="katuosoite"/>
                        <xs:element name="postinumero"/>
                        <xs:element name="postitoimipaikka"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:sequence>
</xs:element>
</xs:schema>

```

6.2 XML-dokumentin suunnittelu

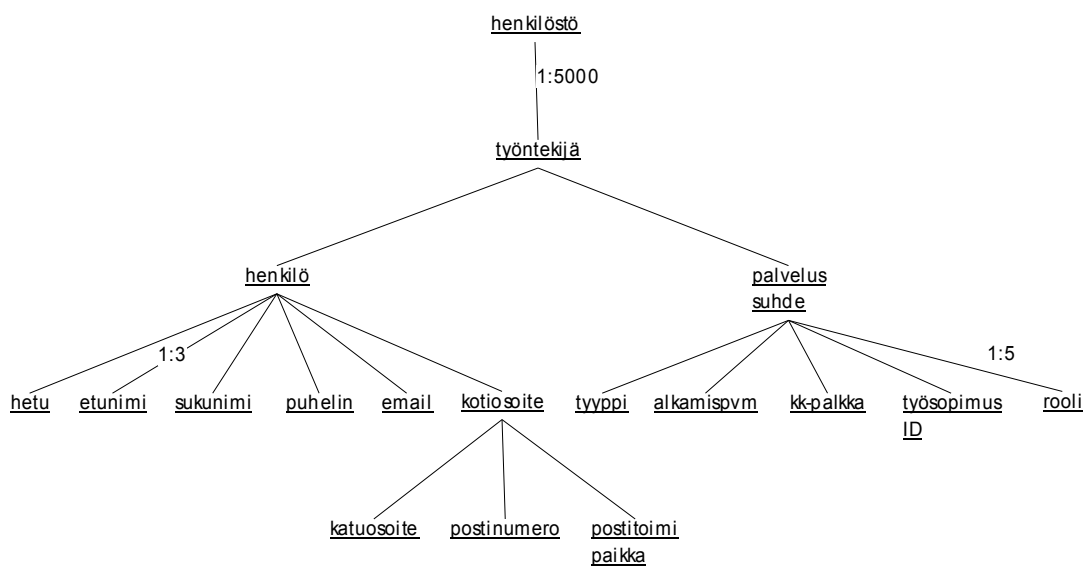
Miten pitäisi määrittää jonkin kohdealueen ilmiötä kuvaavan XML-dokumentin tyyppi tai sen osa, jotta sitä voitaisiin tarvittaessa käyttää järjestelmien välisessä tiedonvaihdossa, asiakirjojen luomisessa, kohdealueen ilmiöiden ja käsitteiden suhteiden kuvaamisessa, tiedon haussa tai vaikkapa monikanavajulkaisemisessa? Toisin sanoen, miten pitäisi nimetä elementit sekä määritellä niiden tyypit ja keskinäiset suhteet.

Nykyään mukaan dokumentin sisältämän merkityksen osoittaminen voidaan hoitaa joko sitä varten toteutetulla ohjelmalla tai käyttämällä dokumentissa etukäteen suunniteltua, nimiavaruuksien avulla dokumenttiin kiinnitettyä sanastoa [32]. Hän toteaa, että suurin osa XML:ään liittyvästä työstä on suunnittelua: mitä merkataan, kenelle ja miksi. Dokumenttien merkkauksessa ei ole järkeä, ellei samalla ole mielessä merkkauksen käyttötarkoitusta

ja sen mukaista sanastoa ja tietorakennetta [32]. Nykäsän mukaan tyypillisiä merkkauksvirheitä ovat mm. merkkauksen valinta esitystavan eikä merkityksen perusteella, saman asian merkkaaminen eri paikoissa eri tavalla sekä asiasisällön ja/tai sanaston merkityksen väärinymmärtäminen.

Barners-Leen mukaan XML on köyhä kieli tiedon mallintamiseen, jos tarkoituksena on esittää kohdealueen informaatioelementtejä siten, että ne vastaavat käyttäjien käsitystä tämän kohdealueen ilmiöistä [1]. Hänen mukaansa ei ole olemassa mitään yleispätevää kehystä uusien merkkauksikielten toteuttamiselle. Niinpä kysymys siitä, mitä XML-dokumentti merkitsee, riippuu ensisijaisesti dokumentin elementtien nimien merkityksistä. Semantiikkaa voidaan liittää jonkin elementin jälkeläiseen vain siinä määrin, kuin on määriteltä kyseessä olevan elementtityypin skeemassa kohteena olevassa nimiavaruudessa. Dokumentin alipuun merkityksestä ei voida puhua ellei ymmärretä kielen semantiikkaa. Koska kielet määrittelevät vain pakolliset merkitykset dokumentille, ainoa tapa voida puhua dokumentin alijoukkojen merkityksestä on määritellä se, miten nuo dokumentin osat voidaan koota uudelleen toiseksi kokonaiseksi dokumentiksi. Jotta kielellä olisi koneellisesti tulkittavissa oleva merkitys, täytyy sen symbolien ja rakenteiden viitata johonkin alla olevaan malliin [1].

Kuva 18. henkilöstö.xml ja henkilöstö.xsd dokumenttien malli



Kuvassa 18 on esitetty malli, jonka pohjalta kuvan 17 skeema-tiedosto on luotu. Malli perustuu käsitteiden intensionaaliseen sisältymissuhteeseen perustuvaan Concept-D kieleen. Käsiterakenne kuvaa erään yrityksen henkilöstön jotakin tarkoitusta varten. Alleviivattuja käsitteitä yhdistävät viivat kertovat, että ylempänä oleva käsite sisältää alempana olevat käsitteet. Viivoilla olevat suhdeluvut kertovat lukumääräsuhteita. Esimerkiksi suhdeluku 1:5000 kertoo, että kyseessä olevan yrityksen enimmäistyöntekijämäärä mallintamishetkellä katsottuna tulee olemaan 5000.

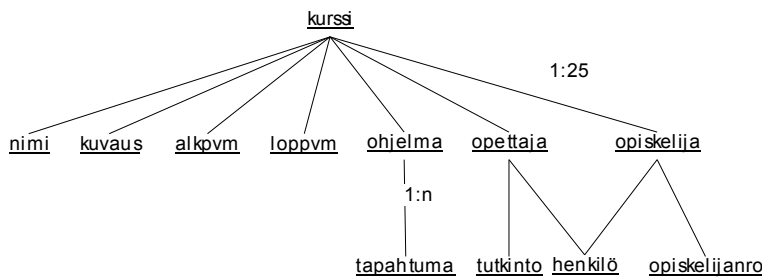
Nykäsen ja Barners-Leen näkemykset XML-dokumenttien merkityksistä tai merkityksettömyyksistä ja mahdollisesta uudelleenkäytöstä antavat ymmärtää, että XML-sinällään ei auta perusongelmien ratkaisemista, jos merkityksien suunnitteluun ei kiinnitetä riittävä huomiota. Näyttäisi siltä, että XML taipuu monipuoliseen käsitteiden kuvaamiseen ja kuvatuilla käsitteillä rakennettujen mallien käsittelyyn. Se ei kuitenkaan ratkaise ongelmia ilman inhimillistä käsitteiden analyysia ja määrittelyä ennen käsitteiden koodaamista koneiden ymmärtämään muotoon.

6.3 Käsiterakenne ja XML

XML-tietorakenne on puumainen. Puun haara- ja lehtisolmut vastaavat XML-tiedoston elementtien nimiä. Jossakin sovelluskohtaisessa XML-nimiavaruudessa määriteltyyn elementtiin voidaan viitata mielivaltaisen monesta muusta nimiavaruudesta, mikäli elementtien merkitys on niissä kaikissa sama. Intensionaaliseen sisältymissuhteeseen perustuva käsiterakenteen kuvaus muodostaa suunnatun syklittömän verkon. Verkkomaisuus johtuu siitä, että kuvattavaa käsitettä määrittelevät käsitteet saattavat sisältää yhden tai useamman yhteisen käsitteen. Mikäli näitä yhteisiä käsitteitä käytetään XML-tiedostoissa elementteinä, näyttäisi XML-tiedoston semantiikan mallintaminen samaistuvan intensionaaliseen käsitteiden mallintamiseen. Tällöin käsiterakennetta esittävä verkko jakautuu useammaksi XML-dokumenttia mallintavaksi puurakenteeksi, joissa yhteisen merkityksen omaavan käsitteen (elementin) tyyppi on määritelty vain kertaalleen. Käsitejärjestelmässä käsitteiden nimien (termien) tulee olla yksikäsitteisiä ts. kahta eri käsitettä ei saa nimetä samalla termillä. XML-dokumentissa voidaan käyttää eri nimiavaruuksissa määriteltyjä elementtejä. Tällöin jollakin kohdealueella käytettävien nimiavaruuksien joukko niissä määriteltyine tyyppineen vastaa Concept-D -kielen jotakin kohdealuetta vastaavaa käsitejärjestelmää. Nimiavaruuksissa määritellyt elementit ja attribuutit, ts. kohdealuetta kuvaava XML-sanasto, voisi olla esimerkiksi kuvassa 10 esitetty käsitekirjasto.

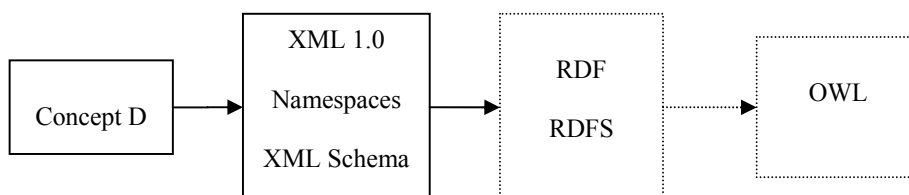
Kuvassa 19 on esimerkki käsiterakenteesta, joka muodostaa syklittömän verkon. Rakente esittää johonkin tarkoitukseen mallinnettua kurssi-käsitettä. Siinä sekä opettaja-käsite että opiskelija-käsite sisältävät intensionaalisesti henkilö-käsitteen. Mikäli henkilö-käsitteen merkitys on kuvassa 19 sama kuin on määritelty kuvan 17 henkilöstö.xsd – dokumentissa, voidaan kurssin tiedot sisältävässä XML-dokumentissa viitata kuvassa 17 esitettyyn nimiavaruuteen. Kuvassa 19 n saa arvokseen kullekin kurssille mahtuvien opiskelijoiden määrän. Mallin pohjalta luotavaa xml-dokumenttia voitaisiin käyttää vaikkapa lomakkeen luomiseksi kurssin www- sivuilla tapahtuvaa ilmoittautumista varten. Mikäli esimerkiksi kurssi-käsite kuvataan alunperin tarkoituksena hallita mallilla kurssin koko elinkaari, voidaan mallista (tyyppitiedostosta) poimia kulloistakin osatarkoitusta vastaava osanen (alipuu). Mikäli käsiterakenne on syntynyt organisaation mallintamisprosessissa, voidaan sitä ja sen sisältämiä käsiterakenteita pitää yhteiskäyttöisinä ainakin organisaation sisällä.

Kuva 19. Kurssi-käsite



Mikäli rakenne edustaa laajemman yhteisön sopimaa tulkintaa, yhteiskäyttöisyys laajenee. Tällöin meillä on yhteinen malli, jonka alipuiden merkitys on kiinnitetty käsiterakenteiden mukaisilla XMK-skeemoilla.

Kuva 20. Käsiterakenteesta ontologiaan

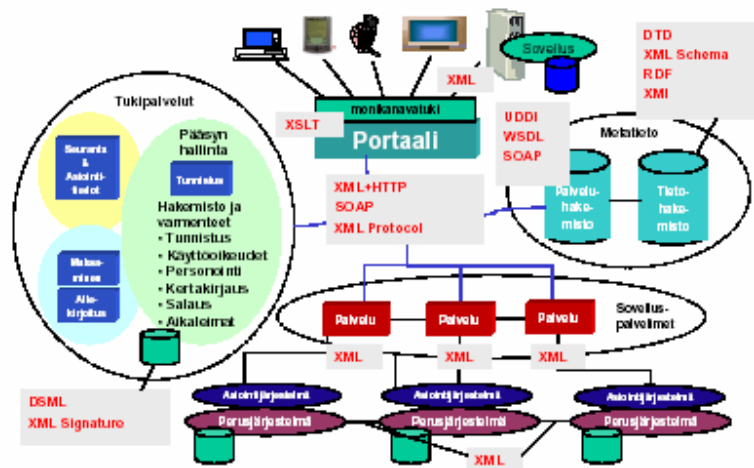


W3C julkaisi kuluvan vuoden helmikuussa kaksi Semanttisen Webin avainteknologi-
aa: uudistetun RDF-kuvailukehyksen sekä Webin ontologiakielen OWL:n. Kuva 20 esittää
mahdollista mallintamisen polkua kohdealueen käsitteiden analyysistä ja mallintamisesta
kohti verkossa käytettävää, hajautetun ontologian hyödyntämistä. Tässä työssä ei ole mah-
dollisuutta paneutua RDF (Revised Resource Description Framework) eikä OWL (Ontolo-
gy Web Language) –suositukseen. Todettakoon vain, että ne perustuvat XML-ytimeen, jolla
kuvattujen dokumenttien mahdollistaakseen XML-tekniikoille asetetut odotukset pitäisi
olla sekä valideja että semanttisesti tarkoitustaan vastaavia.

7 Julkishallinnon sanastotyö

Valtionvarainministeriö antoi vuonna 2001 suositukset valtion tietotekniikan rajapintasuosituksiksi tavoitteena parantaa hallinnon tietojärjestelmien keskinäistä yhteentoimivuutta [38]. Suositus koskee valtionhallinnon lisäksi myös esimerkiksi opetusministeriön alaisia oppilaitoksia. Osana suosituksia määriteltiin myös sähköisen asioinnin looginen tavoitearkkitehtuuri, jossa metatiedoilla ja XML:ään perustuvilla standardeilla on keskeinen rooli. Kuva 21 esittää tavoitearkkitehtuuria ja XML:n roolia siinä. Metatietokokonaisuus koostuu palveluhakemistosta ja tietohakemistosta. Arkkitehtuurin pelkistetty kaavio on esitetty kuvassa 15.

Kuva 21. Tavoitearkkitehtuuri, XML:n rooli [38]



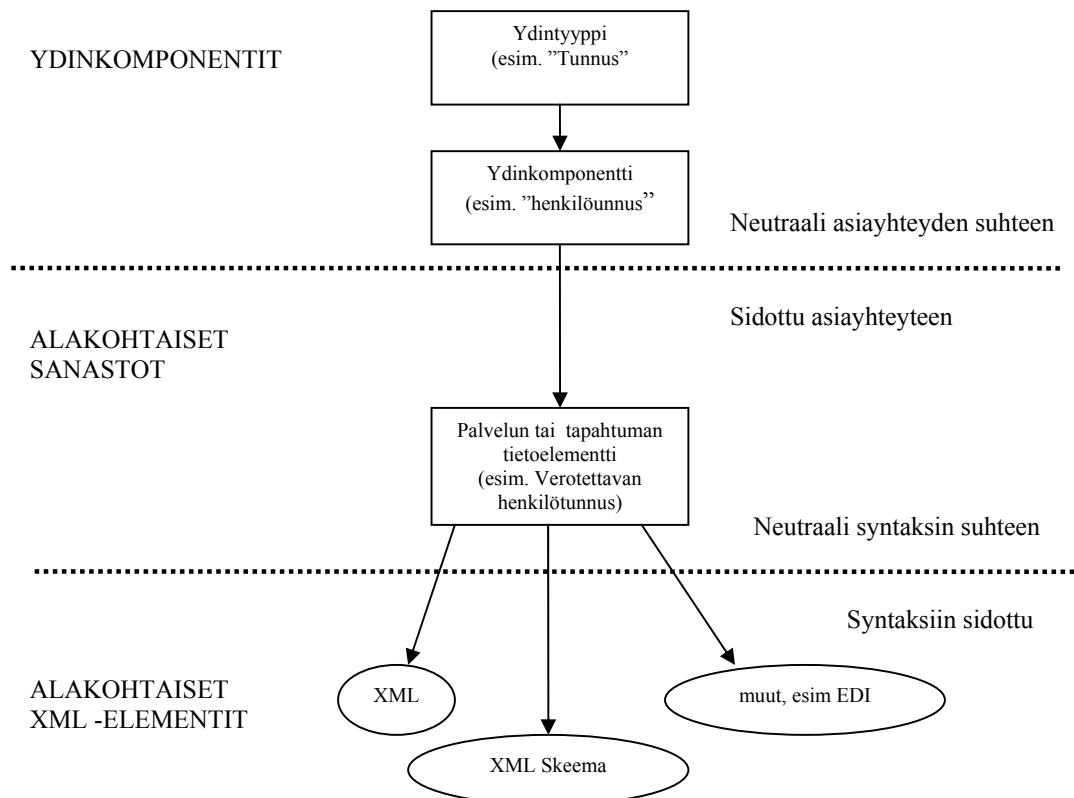
Suosituksen [38] mukaan tietohakemisto voi sisältää sanomakuvauksia, joiden mukaisin sanomin ollaan yhteydessä tiettyyn palveluhakemistossa kuvattuun palveluun. Palveluhakemistojen tarkoitus on tehostaa palvelujen löytymistä esimerkiksi portaalien hakukoneilla etsittäessä. Suosituksessa korostetaan yhteisten sanastojen luomisen merkitystä rakennettaessa yhteistoiminnallisia tietojärjestelmiä palveluketjujen tehostamiseen. Sanastot luokitellaan suosituksessa sovellusaluekohtaisiin, sovellusalueriippumattomiin sekä sanastoviitekehyksiin [38]. Valtiovarainministeriö on julkaissut keväällä 2003 luonnosvaiheessa olevat ohjeet yhteiskäyttöisten sanastojen luomiseksi [39]. Tarkastelen seuraavassa nimeämiskäytännön ohjeistusta.

Sanastolla tarkoitetaan ohjeessa sanastotermeistä koostuvia dokumentoituja luetteloi-
ta, joita käytetään XML rakenteissa käytettävien tietoelementtien luetteluun ja määritte-

lyyn. *XML-sanastolla* tarkoitetaan ohjessa tietyn tyyppisessä dokumentissa mahdollisesti esiintyvien elementtien ja attribuuttien nimiä. Sanastojen kuvaus perustuu UN/CEFACT Core Component -malliin, joka puolestaan perustuu kansainvälisiin standardeihin [39]. Malli perustuu UML-notaatioon.

Kutakin XML-skeemassa käytettyä elementtinimeä vastaa XML-sanastossa ISO 11179-5 standardinmukaisesti nimetty sanastotermin, sen määritelmä ja muuta sen käyttöä koskevaa tietoa [39]. ISO 11179-5 -standardin mukaisesti nimetyn tietoelementin yksilöivä nimi on kolmiosainen ja muodostuu Objektiluokasta (Object Class), Ominaisuustermistä (Property term) ja Ominaisuustermin tietotyyppiin viittaavasta nimestä, joka on ohjeessa suomennettu sanalla 'ilmentymä' (Representation term).

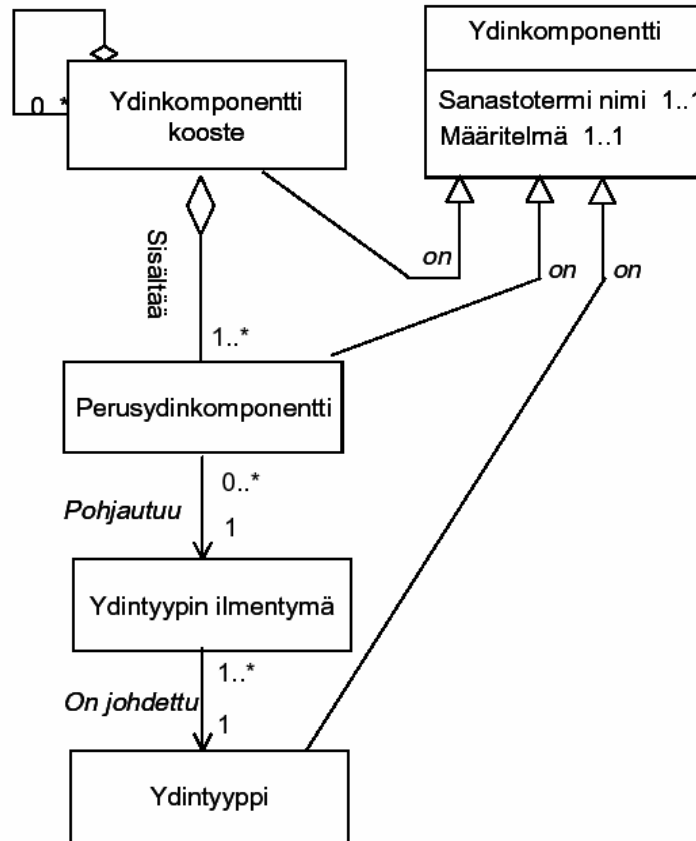
Kuva 22. Sanastojen ja sanastotyön tasot [39]



Kuvassa 22 on esitetty sanastojen ja sanastotyön tasot. Ydinkomponentit ovat neutraaleja sekä syntaksin että asiayhteyden suhteen. Ydinkomponenteista johdetut alakohtaiset sanastot ovat neutraaleja syntaksin suhteen, mutta ne on sidottu asiayhteyteensä. Alakoh-

taisten sanastojen perusteella annetaan nimet johonkin asiayhteyteen sidottaville XML-elementeille.

Kuva 23. Julkishallinnon ohjeistuksen mukainen sanastojen kuvausmalli [39]



Ydinkomponentit määritellään yhdenmukaistetuiksi, asiayhteydestä riippumattomiksi perustason komponenteiksi,

- joita voi sellaisenaan käyttää alakohtaisissa sanastoissa tai
- joista voidaan lisämääreiden avulla johtaa uusia alakohtaisissa sanastoissa tarvittavia täsmällisempiä ilmaisuja ja/tai
- joista voidaan koostaa erilaisia koosteita vastaamaan jonkin alan erityistarpeita.

Ydinkomponentti on yleiskäsite, jolla tarkoitetaan

- perusydydkomponentteja,
- ydintyyppejä tai

- ydinkomponenttikoosteita.

Ohjeessa todetaan edelleen, että Ydinkomponenttikooste on rakenne, joka sisältää yhden tai useampia Perusydinkomponentteja ja/tai Ydinkomponenttikoosteita. Ydintyyppi liittyy tietoelementin nimeen tiedon tietoelementin tietotyypistä. Ydintyyppi koostuu sisältökomponentista ja täydentävästä komponentista. Sisältökomponenttiin voidaan kohdistaa sen tietotyyppiä koskevia rajoittimia. Voidaan esimerkiksi edellyttää, että Mitta on kokonaisluku [39].

Kuvassa 23 on havainnollistettu sanastojen kuvausmallia. Perusydinkomponentti, ydintyyppi ja ydinkomponenttikooste periytyvät ydinkomponentista ts. ydinkomponentti on edellisten ylikuokka. Ydintyyppistä voidaan johtaa ydintyyppien ilmentymiä, joihin mielivaltaisen määrä perusydinkomponentteja voi perustua ja jotka voivat muodostaa ydinkomponenttikoosteita. Perusydinkomponentin malli on kuvattu tarkemmin kuvassa 24.

UN/CEFACT Core Component -spesifikaatiossa on määritelty 10 sallittua ydintyyppiä. Ydinkomponentit ja Ydintyyppit tullaan ohjeen mukaan rekisteröimään julkishallinnossa ydinkomponenttiluetteloon. Siihen dokumentoidaan:

- UID (julkisen hallinnon Ydinkomponentille antama yksikäsitteinen tunniste)
- Nimi
- Ydinkomponentin käyttämä Ydintyyppi
- Määritelmä.

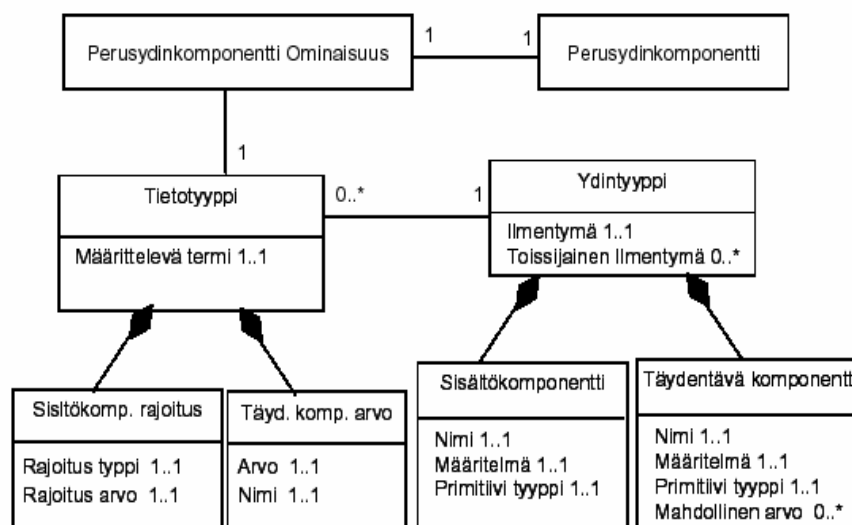
Ydinkomponenttien nimeämisessä käytetään käyttäjäkunnan äidinkielisiä, kirjakielen mukaisia substantiiveja, verbejä ja adjetiiveja niiden perusmuodossa.

Perusydinkomponentti pohjautuu aina johonkin Ydintyyppiin, jolle ydintyyppien ilmentymä antaa nimen. Perusydinkomponentit ovat ydinkomponenttikoosteiden ominaisuuksia ja ovat siksi tiettyä Tietotyyppiä.

Alakohtainen tietoelementti pohjautuu Ydinkomponenttiin ja se määrittellään asiayhteydessään. Asiayhteyttä kuvaavat kategoriat on kuvattu lähemmin UN/CEFACT Core Component spesifikaation luvussa 6.2. Alakohtainen tietoelementti voi olla Perusydinkomponenttia vastaava peruselementti tai Ydinkomponenttikoostetta vastaava tietoelementtikooste.

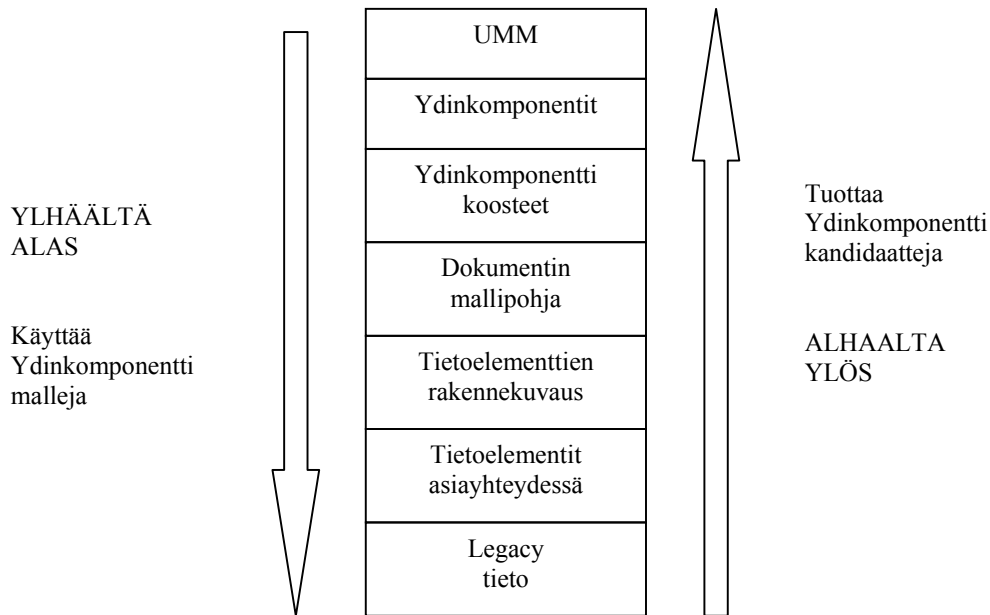
Julkishallinnon toiminta- ja palvelualueiden tietosisällöstä vastaavat tahot ylläpitävät alakohtaisia sanastoja ja XML-rakenteita. Alakohtaiset sanastot pohjautuvat Ydinkomponentteihin sikäli kuin relevantteja ydinkomponentteja on olemassa. Mikäli alakohtaisissa sanastoissa esiintyy yleistä merkitystä omaavia rakenteita, ehdotetaan niitä ydinkomponenttikandidaateiksi. Ohjeessa on kuvattu alakohtaisen sanaston kehittämisen työvaiheet kahdeksana toisiaan seuraavana vaiheena.

Kuva 24 . Perusydinkomponentti [39]



Ohjeessa todetaan: 'Sähköisessä asiointissa eri organisaatioissa ja aloilla käytetyt tietorakenteet vaikka muistuttavatkin monilta osin toisiaan, saattavat sisältää pieniä, mutta merkittäviä semanttisia ja rakenteellisia eroavaisuuksia, joiden taustalla on kulloisenkin käyttöympäristön erityistarpeet. Tämä on aiheuttanut ongelmia tiedon yhteiskäytössä sovelusten ja organisaatoiden välillä' [39].

Kuva 25. Sanaston luontiprosessi [39]



Yhteenveto

UMM malli perustuu UML:n mukaiseen, extensionaaliseen mallinnuksen lähestymistapaan. Ydinkomponenttiluettelot vastaavat sanastoja, joiden termeillä kuvataan XML-dokumenttien elementit ja attribuutit. On sinänsä hyvä asia, että saadaan käyttöön yhteisesti sovitut sanastot. Toisaalta mallissa käytetty terminologia tuntuu sekavalta, joten vaikeuksia toteuttamisessa saattaa ilmetä. Sanastojen luontiprosessissa ei korosteta käsitteistön selkiinnyttämistä tai parantamista. Ohjelunnonksen mukaan prosessissa lähinnä dokumentoidaan olemassaoleva terminologia ja siirretään se sellaisenaan sanastoihin, joiden pohjalta luodaan sovellukset ja niiden käyttämät XML-sanastot.

8. Oppisisältö-käsite ja sen kuvaaminen

LOM, Learning Object Metadata, on suomennettu Tieken tutkimusryhmässä termillä oppisisältö [36]. Myös oppimisaihio-termiä käytetään. Hämeen ammattikorkeakoulussa digitaalisen oppimateriaali -käsitteen kuvaamisen tarkoituksiksi määriteltiin opettajien tarve hallita tuotettavia ja jo tuotettuja sisältökomponentteja siten, että ne ovat helposti löydettävissä ja uudelleenkäytettävissä ja että tekijänoikeudet tulevat huomioituiksi.

Kuvaan tässä luvussa kohdassa 8.1 HAMK:n sisäisiin tarkoituksiin muodostetun oppimateriaalin sisältökomponentti-käsitteen rakenteen. Sisältökomponentilla tarkoitetaan digitaaliseen muotoon talletettua, selaimella käytettävää, jonkin oppiaineen tutkintovaatimuksiin sisältyvää, sisällön mukaan luokiteltavissa olevaa oppimateriaalia. Termillä *sisältökomponentti* viitataan tässä esityksessä jatkossa kuvassa 26 esitettyyn käsiterakenteeseen. Johon on pyritty kokoamaan kaikki ne ominaisuudet, jotka HAMK:ssa on nähty välttämättömiksi komponentin käytön ja sen tekijöiden oikeuksien hallitsemiseksi.

Kohdassa 8.2 esittelen Learning Objects Metadata (LOM) -standardin mukaista oppisisällön metatietojen kuvausta kokonaisuutena. TIEKE:n asiantuntijaryhmä on määritellyt standardin kansallisessa sovellusprofiilissa kuvattavat ominaisuudet [33]. Vertaan sisältökomponentin ominaisuuksia alkuperäisen LOM:n ja LOM:n kansallisen sovellusprofiilin mukaisiin ominaisuuksiin. Vertailun pohjalta linjaan ratkaisun, jossa HAMK:n kokonaistarpeet sisältökomponentin kuvaukselle ja hallinnalle ovat sopuossuussa LOM:n kansallisen profiilin kanssa. Kansallisessa standardin käyttöönotossa suositellut kentät on kuvattu liitteessä 2.

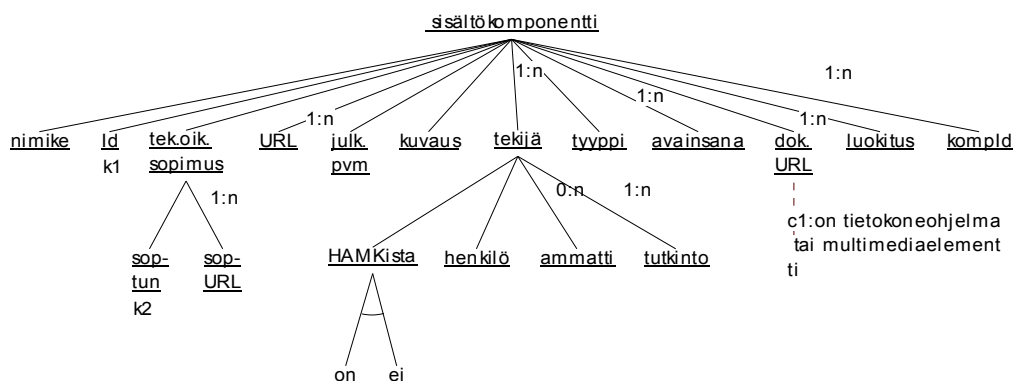
8.1. Sisältökomponentti

Hämeen ammattikorkeakoulussa on määritelty osana digitaalisen opiskelun portaali-projektia digitaalisten oppimateriaalien hallintajärjestelmän vaatimusmäärittelyä. Vaatimusmäärittelyn nykyversiossa todetaan, että selaimella käytettävällä ohjelmistolla pitää voida hallita oppimateriaalien sisältökomponentteja sekä niiden käyttöä. Sisältökomponenttien hallinnalla tarkoitetaan komponenttien systemaattista tallentamista johonkin varastoon siten, että ne löytyvät niiden sisältöön perustuvien hakujen avulla. Sisältökomponenttien käytön hallinnalla taas tarkoitetaan niiden kontrolloitua käyttöä.

Perusideaksi muotoutui alunperin sisältöjen mukaan luokiteltu ja kuvailtu oppimateriaalikomponenttien varasto, josta HAMK:n opettajat voivat valita yksittäisiä oppimateriaalikomponentteja erilaisten ja eri tasoisten opintojaksojen käyttöön. Sisältökomponentin käyttö kirjautuu johonkin järjestelmään siten, että sen perusteella on mahdollista määrittellä tekijälle sopimuksen mukainen korvaus oppimateriaalin käytöstä. Myöhemmin käyttäjäjoukkoa haluttiin laajentaa myös HAMK:n ulkopuolisiin materiaalikomponenttien tarvitsejiin. Sisältökomponentti-käsitteen kuvauksen tarkoitus on siis kuvata ainakin sellaiset piirteet, jotka edesauttavat komponentin löydettävyyttä, päätöksentekoa komponentin käytettävyydestä sekä tekijänoikeuksien turvaamista.

Kuvassa 26 on esitetty sisältökomponentti-käsitteen rakenne kokonaisuudessaan. Liitteessä 1 on kuvattu käsiterakenteeseen sisältyvien muiden käsitteiden sanalliset kuvaukset. Kaikki sisältökomponentti-käsitteen avaimina toimivat ominaisuudet viittaavat HAMK:n omissa järjestelmissä käytettyihin tunnuksiin.

Kuva 26. HAMK:n tarkoituksiin määritelty oppimateriaalin sisältökomponentti-käsite



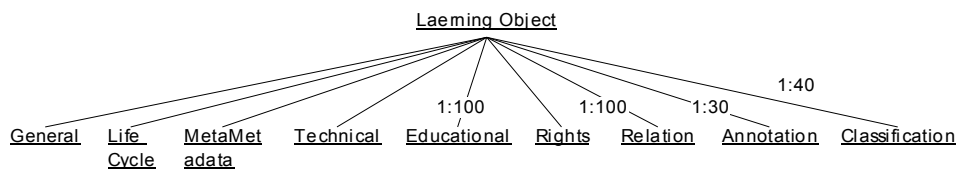
Mallinnuksen oletukset olivat seuraavat:

- sisältökomponentin tekijä tai tekijät voivat kuulua yhtymän henkilökuntaan tai olla ulkopuolisia
- komponentin käytöstä osana oppimateriaalia maksetaan korvaus tekijänoikeussopimuksessa määritettyjen ehtojen mukaisesti.

8.2 Oppisisällön ja sisältökomponentin vertailu

LOM standardin suomennoksessa todetaan, että standardi kuvaa oppisisällön käsitteellisen tietorakenteen, jossa oleelliset ominaisuudet on ryhmitelty yhdeksään luokkaan. Standardissa käytetään myös termiä metatietorakenne. Luokalla tarkoitetaan standardissa toisiinsa jossakin suhteessa olevien tietoalkioiden ryhmää. Standardissa oppisisällöllä tarkoitetaan mitä tahansa digitaalista tai ei-digitaalista tieto-oliota, jota voidaan käyttää oppimiseen, opettamiseen tai kouluttamiseen. 27 esittää standardin mukaista oppisisällön ominaisuuksien luokitusta. LOM-standardi kokonaisuutena on esitetty liitteessä 3, tummenneet elementit kuvaavat kansalliseen sovellusprofiiliin valittuja elementtejä.

Kuva 27. LOM-standardin perusluokitus.



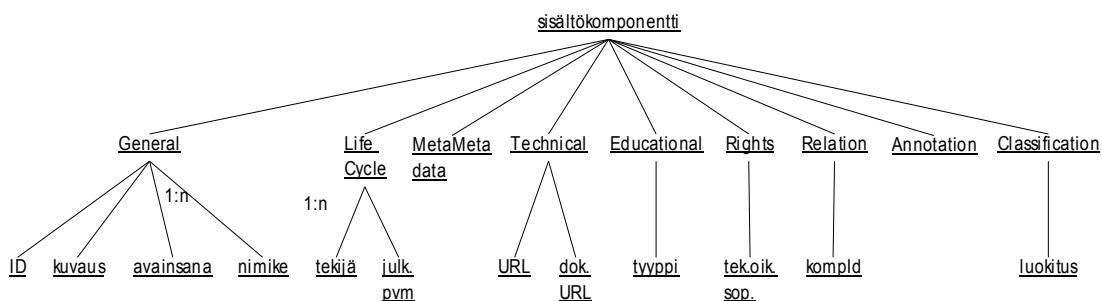
Standardissa todetaan, että perusrakenteeseen tehtävien laajennosten tulee säilyttää LOMv1.0:n puun lehtisolmuille kuvattu tietoalkioiden arvoavaruus ja tietotyyppi. Semantiikan mahdollistaman vuorovaikutteisuuden säilymiseksi mukautetussa mallissa ei saa korvata alkuperäistä LOM tietoalkiota, mutta sekä yksittäistä eikä aggregoimalla muodostettua ominaisuutta voidaan laajentaa lisäämällä sille ominaisuuden arvoavaruuden ja tietotyypin säilyttäviä lapsielementtejä.

Kuvaan seuraavassa perusluokituksen ja kunkin luokan kohdalla kotimaisessa sovellusprofiilissa [tie] kuvattavien ominaisuuksien piirteet. Vertaan kuvauksen yhteydessä LOM:ssa kuvattuja ominaisuuksia sisältökomponentin ominaisuuksiin ja muodostan synteesin, jossa toteutuvat sekä sisältökomponentin että oppisisällön kansallisen suosituksen mukaiset ominaisuudet. Standardin käytön kannalta on oleellista, että ominaisuuksia edustavat termit vastaavat alkuperäisiä suosituksia ts. tieto-alkioita edustavat merkkijonot säilyvät alkuperäisinä. Vertailun helpottamiseksi olen luokitellut sisältökomponentin ominaisuudet LOM:n ominaisuusluokituksen mukaiseksi, kuva 28.

1. General (Yleinen) luokka

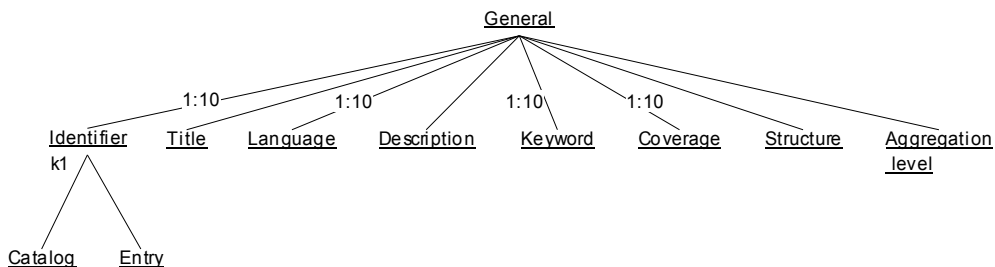
General-luokkaan on ryhmitelty oppisisältöä kokonaisuutena kuvaava yleinen informaatio. Kansallisissa oppisisältökuvauksissa pakollisia kuvattavia ominaisuuksia ovat Language (kieli)- ja Description (kuvaus)-ominaisuudet. Standardissa todetaan, että Identifier (tunniste)-ominaisuuden arvo identifioi globaalisti yksikäsitteisesti oppisisällön. Tämä ei kuitenkaan päde digitaalisiin oppisisältöihin. Identiteetti-kriteerin pitää olla jokin pysyvä ominaisuus. Verkko-osoitetta ei voida pitää sellaisena. Koska standardin mukaan alkuperäistä LOM-tietoalkioita ei saa korvata, mutta sitä voidaan laajentaa, lisätään Identifier-ominaisuuden kentät HAMK-hakemisto ja HAMK-tunnus.

Kuva 28. Sisältökomponentin ominaispiirteet/LOM:n perusluokitus



Keyword (avainsana)-ominaisuus on pakollinen, mikäli kuvauksessa ei käytetä Classification (luokitus)-luokassa kuvattavia luokittelusysteemejä. Kuvassa 29 on esitetty alkuperäisen LOM metatietorakenteen kaikki luokkaan General määritellyt ominaisuudet.

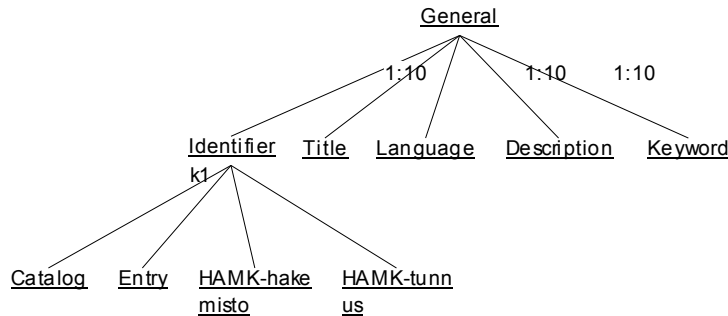
Kuva 29. General-kategoriassa kuvattavat ominaisuudet (LOM oppisisältö).



Liitteessä 2a on kuvattu kansallisen suosituksen mukaiset General-luokkaan kuuluvat ominaisuudet. Taulukossa 3 on verrattu sisältökomponentin General-luokan ominaisuuksia

vastaavia tunnusmerkkejä ja LOM-standardin vastaavan luokan ominaisuuksia keskenään. Kuva 30 esittää LOM:n kansallisen sovellusprofiilin mukaista oppimisaihion ja sisältökomponentin synteesiä. General-luokkaan valitut käsitteet on kuvattu taulukossa 3 Title-ominaisuutta lukuun ottamatta. Title-ominaisuus vastaa kuvassa 28 esitettyä sisältökomponenttiin sisältyvää nimike-ominaisuutta.

Kuva 30. Sisältökomponentin ja LOMv1.0:n General-luokan synteesi.



Taulukko 3. Sisältökomponentin ja LOM:n General-luokan ominaisuuksien vertailu.

Sisältökomponentti	Oppisisältö (LOM)	Vertailun tulos	Synteesi/valinta
Id	Identifier. Catalog Identifier. Entry	<u>Identifier</u> ominaisuus on laajempi kuin <u>tunnus</u> -ominaisuus, mutta ei identifioi yksikäsitteisesti oppimisaihiota. Laajennetaan kentillä HAMK-hakemisto ja HAMK-tunnus	Identifier. Catalog Identifier. Entry HAMK-hakemisto HAMK-Id
kuvaus	Description	vastaavat toisiaan	Description
avainsana	Keyword	vastaavat toisiaan	Keyword
-	Language	ominaisuus puuttuu sisältökomponentista	Language
nimike	Title	vastaavat toisiaan, ei pakollinen LOM:n kansallisessa profiilissa, mutta pakollinen sisältökomponentissa	Title

2. Lifecycle (Elinkaari) luokka

Lifecycle-kategoriaan ryhmitellään ominaisuudet, jotka kuvaavat oppisisällön historiaa ja nykyistä tilaa sekä oppisisältöön sen kehityksen aikana vaikuttaneiden henkilöiden ja organisaatioiden tietoja. Kuvassa 31 on esitetty luokassa kuvattavat ominaisuudet. Suomessa pakollisiksi oppisisällön elinkaarta kuvaaviksi ominaisuuksiksi on määritelty oppisisällön tilaa kuvaava Status- sekä Contribute-ominaisuudet. Koska Version-ominaisuutta ei vaadita kansallisessa sovellusprofiilissa eikä se sisälly HAMK:ssa määriteltyyn sisältökomponenttikäsitteeseen, jätetään ominaisuus kuvauksesta pois.

Sisältökomponentti-käsitteeseen sisältyy tekijä-käsite. Tekijä-elementti sisältyy LOM:n Lifecycle-luokan Contribute-ominaisuuden arvojoukkoon. Toisaalta sisältökomponen-

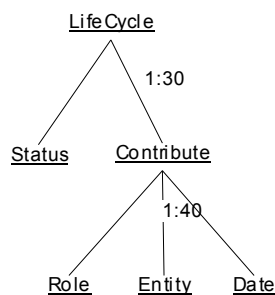
tin tekijä-käsite sisältää intensionaalisesti käsitteen `Contribute.Entity`. Sisältökomponentin tekijä-käsite sisältää enemmän tunnusmerkkejä ja siten sillä voidaan ilmaista enemmän informaatiota kuin `LifeCycle`-luokan `Contribute`-ominaisuudella.

Oppisisältöön vaikuttaneiden tahojen tiedot ilmaistaan IMC vCard 3.0 [19] määrittelyllä. Sosiaaliturvatunnusta (sotu) ja postiosoitetta lukuun ottamatta vCard:n sisältö on sama kuin sisältökomponentin tekijä.henkilö-käsitteellä. Sisältökomponentti käsitteen määrittelyllä, kuva 26, pyrittiin kuvaamaan kaikki HAMK:n tarkoituksiin vaadittavat sisältökomponentin ominaisuudet. Se ei kuitenkaan tarkoita sitä, että kaikkia ominaisuuksia pitäisi hallita sisältökomponentin metatiedon tallennusjärjestelmässä. Täten kansallisen sovellusprofiilin mukainen `LifeCycle`-luokan ominaisuuksien kuvaus riittää HAMK:n tarkoituksiin kuvattaessa pelkkiä metatietoja. Tekijänoikeuksien takaamiseksi tarvitaan erillinen järjestelmä. Sen kuvaaminen ei ole tämän työn tehtävä.

Taulukko 4. Sisältökomponentin ja LOM:n `LifeCycle`-luokan ominaisuuksien vertailu.

Sisältökomponentti	Oppisisältö (LOM)	Vertailun tulos	Synteesi/valinta
-	Status	puuttuu sisältökomponentista	Status
tekijä	Contribute.Role	Sisältökomponentissa suppeampi, <code>Contribute.Role.author=tekijä</code>	Contribute.Role
tekijä.henkilö	Contribute.Entity	vastaavat toisiaan	Contribute.Entity
julk.pvm	Date	vastaavat toisiaa	Date

Kuva 31. Lifecycle kategorian käsitteellinen tietorakenne



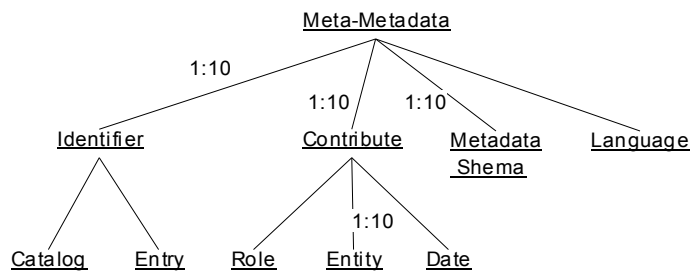
Kuva 31 esittää kansallisen sovellusprofiilin mukaista `LifeCycle`-luokan ominaisuuksien määrittelyä.

3. Meta-Metadatan (Tieto tiedosta) luokka

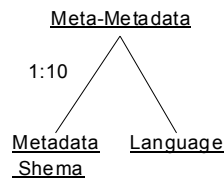
Meta-Metadatan -luokkaan sisältyy informaatio oppisisältöä kuvaavan metatiedon ilmentymästä itsestään (kuva 32). Suomalaisissa kuvauksissa pakollisiksi on määritelty Metadata Schema ja Language -kentissä määriteltävät tiedot.

Sisältökomponentti-käsitteeseen ei sisälly Meta-Metadatan -luokan ominaisuuksia. Kuva 33 esittää liitteessä 2b kuvattuja kansallisen profiilin mukaan kuvattavia luokan ominaisuuksia.

Kuva 32. LOM:n Meta-Metadatan luokka



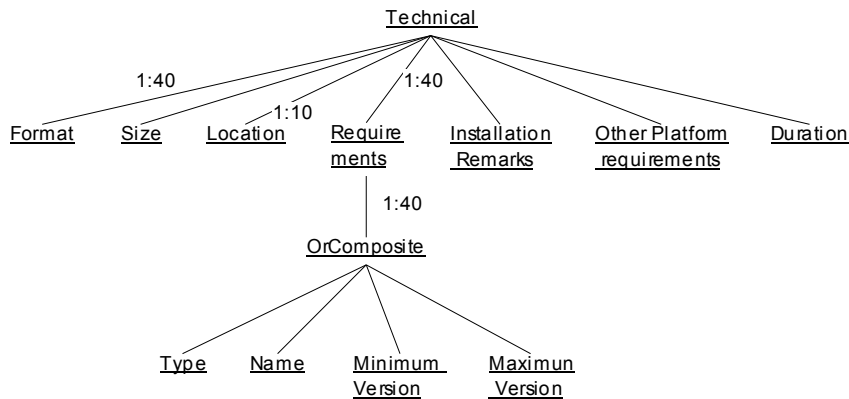
Kuva 33. Meta-Metadatan luokan rakenne synteessin jälkeen.



4. Technical (Tekniset ominaisuudet) luokka

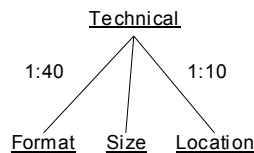
Technical (Tekniset ominaisuudet) kategoriaan luokitellaan oppisisällön teknisiä ominaisuuksia ja niiden edellyttämiä laitteisto- ja ohjelmistovaatimuksia kuvaavat piirteet, kuva 34. Kotimaisissa kuvauksissa pakolliset ominaisuudet ovat Format-, Size- ja Location-ominaisuudet.

Kuva 34. Technical-luokkaan kuuluvat ominaisuudet



Sisältökomponentin URL-ominaisuus kuuluu Technical-luokassa kuvattaviin ominaisuuksiin. Niiden lisäksi kansallisessa profiilissa ilmoitetaan myös tiedoston koko (Size) sekä Format-ominaisuus, joka ilmaisee oppisisällön teknisen tietotyypin tai tekniset tietotyypit. Kuva 35 kuvaa luokan kansallisen sovellusprofiilin mukaiset ominaisuudet (Liite 2b).

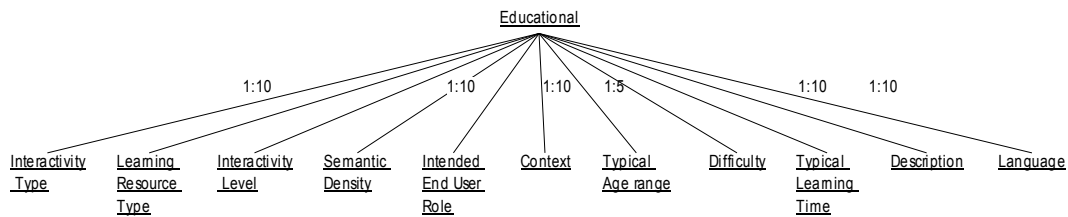
Kuva 35. Technical luokka synteesin jälkeen



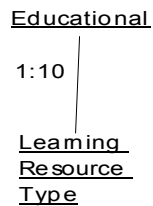
5. Educational (Opetukselliset ominaisuudet)

Educational kategoriaan ryhmitellyt tietoalkiot kuvaavat oppisisällön opetukselliset ja pedagogiset ominaisuudet. Suomessa ainoa pakollinen tämän luokan kuvattava ominaisuus on Learning Resource Type, joka vastaa sisältökomponentin tyyppi-ominaisuutta (Liite 2c).

Kuva 36. Educational-kategorian ominaisuudet



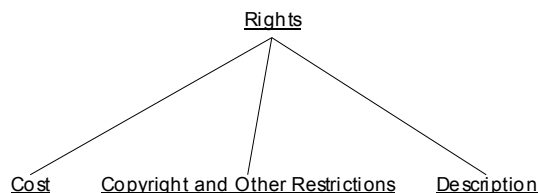
Kuva 37. Educational –luokka synteessin jälkeen.



6. Rights (Oikeudet) luokka

Rights-kategoriaan kuuluvat oppisisällön tekijänoikeuksia ja käyttöehtoja koskevat ominaisuudet. Luokan ominaisuudet kuvataan kaikki myös kansallisessa sovellusprofiilissa, kuva 38.

Kuva 38. Rights-kategoriassa kuvattavat ominaisuudet.



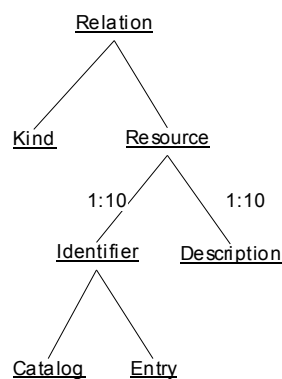
Sisältökomponentti-käsitteen sisältämään tekijänoikeussopimus-käsitteeseen sisältyvät sopimusTunnus- ja sopimusURL-käsitteet. LOM:n mukainen Rights-luokka ei sisällä näitä tietoja. Kuten LifeCycle-luokan käsittelyn yhteydessä totesin, sisältökomponentti-käsitteen määrittelyn kuvaustarkoitus on ollut laajempi kuin LOM:n kuvauksen tarkoitus. Ei ole syytä laajentaa Rights-luokkaa sisältökomponentin tekijänoikeuksia koskevilla ominaisuuksilla, koska tekijänoikeuksiin liittyvät tiedot ja toiminnot tullaan hallinnoimaan sisällön kuvailusta erillisessä järjestelmässä. Liitteessä 2c on kuvattu alkuperäisen LOM-standardin mukai-

set Rights-luokan ominaisuudet, jotka vastaavat myös kansallista sovellusprofiilia ja HAM-Kin tarpeita.

7. Relation (Suhteet) -luokka

Relation-kategoriaan kuuluvat piirteet, jotka määrittävät oppisisällön ja siihen liittyvien muiden oppisisältöjen keskinäiset suhteet. Ominaisuudet ja niiden suhteet on esitetty kuvassa 39. Tämän luokan ominaisuuksia ei ole määritelty pakollisiksi suomalaisessa versiossa.

Kuva 39. Relation luokan ominaisuudet.

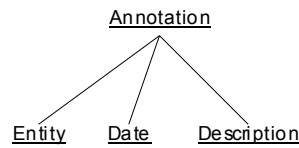


Kansallinen profiili ei edellytä Relation-luokan ominaisuuksien kuvaamista. Sisältökomponentin määritelmän mukaan komponentti voi sisältää myös muita komponentteja. Liitteessä 2c on kuvattu Relation-luokan ne ominaisuudet, jotka ovat välttämättömiä kuvaamaan sisältökomponentin sisällymistä toiseen komponenttiin. Description-ominaisuus ei ole edellämämainitussa mielessä välttämätön, mutta lisää tietoa komponentin käytettävyydestä.

8. Annotation (Huomautukset) -luokka

Annotation-kategoriaan kuuluvat oppisisällön opetukselliseen käyttöön liittyvät kommentit, tiedot niiden tuottajista ja ajankohdista. Kuvassa 40 kuvatut ominaisuudet eivät ole pakollisia suomalaisessa LOM:n mukaisen oppisisältökuvauksen käyttöönotossa eikä niitä sisälly sisältökomponenttiin. Täten koko luokka jätetään kuvauksesta pois.

Kuva 40. Annotation luokan ominaisuudet.

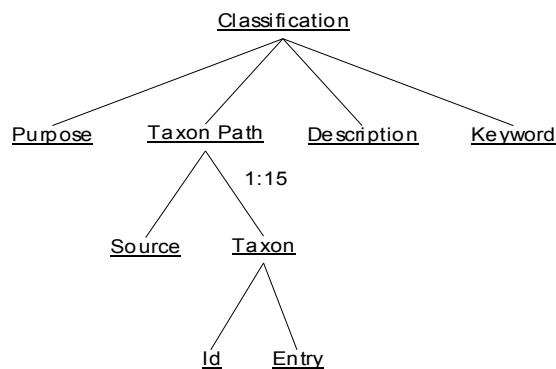


9. Classification (Luokitus) -luokka

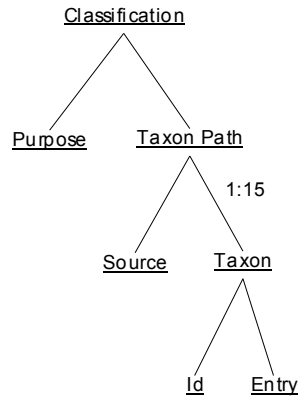
Classification-kategoriaan kuuluvat ominaisuudet kuvaavat oppisisällön suhteet ennaltamääritelyihin luokittelujärjestelmiin, jotka voivat olla nk. virallisia luokituksia tai käyttäjien määrittelemiä omia, kohdealueella käytettäviä luokituksia. Mikäli luokassa General ei ole määritelty avainsanoja (Keywords), on tämän luokan ominaisuudet Description ja Keywords ominaisuuksia lukuun ottamatta pakollisia käytettäessä LOM-standardin mukaista kuvausta kansallisissa oppisisältö-kuvauksissa. Classification-luokan ominaisuudet ja niiden väliset suhteet on esitetty kuvassa 41.

Classification-luokan ominaisuudet on määritelty vaihtoehtoisiksi kuvattaviksi General.Keywords-ominaisuuden kanssa. Sisältökomponentti-käsitteeseen sisältyivät sekä luokitus-käsite että avainsanat. Kuva 42 esittää Classification-luokan kansallisen sovellusprofiilin mukaisia ominaisuuksia synteesin jälkeen. Ominaisuudet on kuvattu liitteessä 2d.

Kuva 41. Classification luokkaan luokitellut ominaisuudet.



Kuva 42. Classification luokan kansallisessa versiossa kuvattavat ominaisuudet.



8.3 Yhteenveto

Sisältökomponentti-käsitteen kuvauksen tarkoitus oli sisällyttää käsiterakenteeseen kaikki sellaiset tunnusmerkit, joiden avulla pystytään valitsemaan komponentti osaksi jonkun määrätyn opintojakson digitaalista oppimateriaalia sekä kontrolloimaan komponentin käyttöä. Käytön kontrollin tarkoituksena on turvata komponenttiin liittyvien tekijänoikeuksien suojaaminen. Tekijänoikeuksien omistaja voi olla joku muukin kuin tekijä. Koska tekijän ammattitaitoa kuvaavat ominaisuudet – tutkinto ja tehtävänimike – ovat yksi kriteeri komponentin laadusta ja siten helpottavat komponentin valitsijan päätöksentekoa, sisällytettiin ne HAMK:ssa sisältökomponentin ominaispiirteisiin. Komponentin käytön kontrolliin liittyvät ominaisuudet puuttuvat LOM-standardin mukaisesta oppisisällön metatietorakennekuvauksesta. LOM-standardin tarkoitus on helpottaa oppijoiden, opettajien ja tietokoneohjelmien oppisisältöön kohdistuvaa etsimistä, arviointia, hankkimista ja käyttöä. Tässä käytöllä ei kuitenkaan tarkoiteta käytön kontrollointia. Täten siihen vaadittavia ominaisuuksia ei ole sisällytetty oppisisällön standardin metatietorakenteeseen.

LOM-standardin mukainen metatietomäärittely vastaa myös HAMK:n tarpeita siinä tapauksessa, että komponentin tekijänoikeuksia ja käyttöä hallitaan muussa kuin metatietojen talletussysteemissä.

9 Johtopäätökset

XML-muotoinen merkintäkieli määrittelee tiedon rakenteen syntaksin, mutta tiedon semanttisesta merkityksestä se ei kerro mitään. Eri päätelaitteiden kautta tapahtuva sähköinen asiointi ja sen edellytyksen muodostava perusjärjestelmien keskinäinen kommunikointi helpottuvat, mikäli määritellään ja otetaan käyttöön kullekin organisaatiolle ja suuremmissa mittakaavassa kansallisille ja kansainvälisille sovellusalueille yhteiset, yhteisesti määritellyt sanastot. Nämä sanastot voidaan kuvata XML metakielellä muodostetuilla merkkauksielillä. Jotta sanastojen sisältämät termit ymmärrettäisiin yhtenevällä tavalla, pitää termien merkityksen määräävien käsitteiden rakenne, merkitys ja keskinäiset suhteet analysoida ja kuvata. Eri tahojen sitoutuminen näiden kuvausten käyttöön vasta mahdollistaa ko. tahojen järjestelmien joustavan yhteistoiminnallisuuden.

Jotta sekä ihmiset, koneet ja ohjelmistot ymmärtäisivät samaa kieltä, pitää käsitteiden merkitykset kuvata kaikkien osapuolien ymmärtämällä tavalla. Concept-D:n peruslogiikka ja visuaalinen esittämistapa ovat helposti ihmisen ymmärrettävissä. Kuvauskielellä voidaan määritellä käsiterakenteet täsmällisesti verrattuna luonnollisella kielellä tehtyihin määritelmiin. Ennenkuin käsitteitä voidaan kuvata XML-merkintäkielellä, on hankittava tietoa ja tietämystä kohdealueesta muodossa, jota myös kohdealueella työskentelevät ihmiset ymmärtävät. Tietämyksen hankkiminen ja analysointi ts. kohdealueen mallintaminen on edellytys kohdealueen ontologian rakentamiselle. Analysoitujen ja yhteisesti sovittujen käsitteiden koodaaminen XML-dokumenteiksi mahdollistaa semanttisen Webin standardien täysipainoisen hyödyntämisen.

10 Jatkotutkimuksen aiheita

Käsitteellinen mallintaminen kattaa erittäin laajan alueen. Tässä esityksessä on ollut mahdollista tutkia sitä vain pintapuolisesti, lähinnä terminologiaa selvitellessä. Miltei jokaisesta kappaleesta löytyisi mielenkiintoisia jatkotutkimuksen aiheita. Kolme aiheita nousevat kuitenkin ylitse muiden.

10.1 Concept-D:n käyttö XML-dokumenttien ja sanastojen mallinnuksessa

Näyttäisi siltä, että Concept-D:n logiikka ja kuvaustapa tukisivat myös monimutkaisempia käsiterakenteita kuin kohdassa 6.5 olen lähinnä intuition pohjalta esittänyt. Tämä edellyttäisi

1. jonkin konkreettisen, hyvin rajatun mallintamistehtävän esittämistä ja sen kuvaamista Concept-D –kielellä
2. sen tutkimista, voidaanko määritellyt käsiterakenteet kuvata XML-skeemalla ja jos ei voida, niin missä tulee raja vastaan. Ts. mitä Concept-D –kielessä määriteltyjä käsitteiden ominaispiirteitä ja minkä tyyppisten käsitteiden ominaispiirteitä ei voida kuvata XML-skeemalla.

Jos kohdassa 2 saataisiin lupaavia tuloksia, pitäisi ne vielä empiirisesti todeta oikeiksi eli rakentaa prototyypisovellus mallintamistehtävän mukaisesta kohdealueesta riittävän laajalla aineistolla. Mikäli prototyypin kokeilu tuottaisi lupaavia tuloksia, seuraava vaihe XML-dokumenttien suunnittelun systemaattisessa kehittämisessä olisi kehittää editori, joka suoraan muuntaisi Concept-D –kielellä kuvatut käsiterakenteet valideiksi XML-dokumenteiksi.

10.2 Julkishallinnon sanastotyön tutkiminen

Jos julkishallinnon yhteiset sanastot tulkitaan vain käytettyjen termien luetteloiksi, yhteistoiminnallisuuden rakentaminen eri sektoreiden sisällä ja välillä, tulee aiheuttamaan runsaasti lisämäärittelyjä. Erittäin mielenkiintoinen tutkimuskohde olisi julkishallinnon sanastoje analysointi OntoClean-metodin pohjalta.

10.3 LOM ja käytännön sovellukset

LOM-standardin julkistamisesta on kulunut kohta kaksi vuotta. Suomessa kehitys asian tiimoilta on miltei pysähtynyt varojen puutteeseen. Verkko-opetuksen lisäämisen tär-

keyttä ja merkitystä korostetaan kuitenkin voimakkaasti opetusministeriön ja opetushallinnon suunnalta. Kaupallisia, puhtaasti LOM:aan perustuvia sovelluksia ei ole tullut markkinoille. OpenSource-sovelluksia lienee kehitteillä. Ts. olisi mielenkiintoista tutkia, mikä merkitys LOM-standardilla on ja tulee mahdollisesti olemaan oppisisältöjen metatietojen kuvauksissa. Onko metatietomalli liian laaja ja vaativa toteutettavaksi?

Lähteet

- [1] Barners-Lee T., The Semantic Web, <http://www.inf.puc-rio.br/~casanova/Topicos-WebBD-Referencias/Outros/Barners-Lee-Scientific-American-May-2001.PDF>. Viitattu 15.6.2003.
- [2] Bird L., Goodchild A., Halpin T., Object Role Modelling and XML -Schema. <http://www.orm.net>. Viitattu 15.6.2003
- [3] Boman, M., Bubenko Jr, J. A., Johannesson, P. & Wangler, B. Conceptual modelling. Prentice Hall, Great Britai, 1997.
- [4] Calvanese D., De Giacomo G., Lenzerini M., Nardi D., Rosati R., Information Integration: Conceptual modeling and reasoning support. Proceedings Of the 6th International Conference on Cooperative Information Systems (CoopIS-97),1997.
- [5] Chandrasekaran, B., Josephson, J.R. and Benjamins, V.R., What are ontologies, and why do we need them?, IEEE Intelligent Systems, January/February 1999. <http://xml.coverpages.org/XMLAndSemantics.html/>. Viitattu 18.2.2003.
- [6] Duineveld A. J., Stoter R., Weiden M. R., Keneba B., Benjamins V. R. , WonderTools? A comparative study of ontological engineering tools. <http://www.swi.psy.uva.nl/wondertools/>. Viitattu 2.7. 2003.
- [7] Elmasri R., Navathe S. B., Fundamentals of Database Systems. 3rd edition. Addition-Wesley, 2000.
- [8] Eysenck M.W., The Blackwell Dictionary of Cognitive Psychology. Blackwell Reference, 1990.
- [9] Frisby J.P., Roth I., Perception and Representation. A Cognitive Approach. Open University Press, Milton Keynes, England 1976
- [10] Gangeni A., Guarino N., Oltramari A., Conceptual Analysis of Lexical Taxonomies: The Case of WordNet Toplevel.,<http://www.loa-cnr.it/Papers/FOIS2001-Final.pdf>. Viitattu 10.9.2004
- [11] Gruber, T.,A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993. ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-92-71.ps
- [12] Guarino N., Welty C., Supporting Ontological Analysis of Taconomic Relationship. Data and Knowledge Engineering39(1), pp51-74. LADSEB-CNR Technical Report 11/2001.
- [13] Guarino N., Formal Ontology and Information Systems. Proceedings of FOIS'97, Trento, Italy 1997. Amsterdam, IOS Press.
- [14] Guarino N., Ontologies and Knowledge Bases: Towards a terminological Clarification. National Research Council, LADSEB-CNR, Padova,Italy.
- [15] Guarino N., Welty C., Formal Ontology of Properties. Ed. Rose Dieng. Proceedings of 12th International Conference on Knowledge Engineering and Knowledge management. Lecture notes on Computer Science. Springer Verlag 2000.
- [16] Guarino, N., Welty C., Ontological Analysis of Taxonomic Relationships. Proceedings of ER-2000: The International Conference on Conceptual Modeling. October, 2000. Springer-Verlag LNCS.
- [17] Guarino N., Welty C., Evaluating Ontological Decisions with OntoClean. Communications of the ACM. February 2002/Vol.45. No 2. Sivut 61-65.
- [18] Helsingin yliopiston kirjasto, <http://www.lib.helsinki.fi/kirjastoala/asiasanastot/ysa.htm>. Viitattu 15.1.2004
- [19] IMC, IMC vCard, <http://www.imc.org/pdi/>. viitattu 10.10.2003
- [20] Museovirasto, Kamut-2, Muistorganisaatioiden tietovarannot yhteiskäyttöön. Toteuttamismahdollisuudet ja toimenpidesuositus. Kamut -2 yhteishankkeen loppuraportti 16.1.2004. <http://www.nba.fi/tiedostot/8c050dc5.pdf>. viitattu 3.4.2004
- [21] Kangassalo H., COMIC – A System and Methodology for Conceptual Modelling and Information Construction. Data&Knowledge Engineering 9(1992/1993) pp.277-319.
- [22] Kangassalo H., Kuvauskieli käsitteellistä mallintamista varten, CONCEPT-D. Tietojenkäsittelyopin laitos, Tampereen yliopisto, Tampere, 1990.
- [23] Kangassalo H., Frameworks of Information Modelling: Construction of Concepts and Knowledge by Using the Intensional Approach. Springer-Verlag Berlin Heidelberg 2000.

- [24] Kangassalo H., Käsitteellinen mallintaminen. Luennot. Tampereen yliopisto. Syksy 2001.
- [25] Kauppi R., Einführung in die Theorie der Begriffssysteme. Tampereen yliopisto. Tampere 1967.
- [26] Kutvonen M., CWM standardoi metadatan. Sytyke ry – Systeemityö 3/03, TTL.
- [27] Lopez F., Overview of Methodologies for building ontologies, Proc. of IJCAI-99, workshop KRR5, Sweden, 1999.
- [28] Lutz M.: Can the Vision of Reusable Ontologies be put into Practice? GIScience 2002, Booklet of Extended Abstracts, Boulder.
- [29] Meersman R., Stanford DB Seminar 991112 Data semantics, modeling and ontologies: New frontiers in databases. <http://www-db.stanford.edu/dbseminar/Archive/Fally99/meersman-slides/sld006.htm>. Viitattu 10.10.2003
- [30] Niemi T., CoSMoS, Conceptual Schema Modelling System, <http://www.cs.uta.fi/~tapio/cosmos/cosmoswebhelp/index.html>. Viitattu 3.3.2002
- [31] Noy N. F., McGuinness D. L., Ontology Development 101: A Guide to Creating your First Ontology. <http://serv-4100.dfki.uni-kl.de:8000/~docbase/dokana/WWW/D00000606.pdf>. Viitattu 10.6.2003.
- [32] Nykänen O., XML, Docendo Finland Oy, 2003.
- [33] Tieke, LOM, kansallinen käyttöönotto. <http://www.tieke.fi/standardointi.nsf/>. Viitattu 9.10.2003
- [34] Salminen A., Towards Digital Government by XML Standardization: Methods and Experiences. Proceedings in Conference Open standards, XML and the Public Sector. Sivut 5-15. October, 2003. XML Finland.
- [35] Smith B., Ontology and Information Systems. <http://ontology.buffalo.edu/ontology%28PIC%29.pdf>. Viitattu 10.10.2002.
- [36] Tieke, LOM Oppisisällön metatieto, IEEE 1484.12.1-2002. Suomennosluonnos Jukka Korpela, [http://www.tieke.fi/standardointi.nsf/38e4483ea7238da4c225650f004a738d/66001655e78fb82cc2256bd80045d9c8/\\$FILE/LOM_fi.html](http://www.tieke.fi/standardointi.nsf/38e4483ea7238da4c225650f004a738d/66001655e78fb82cc2256bd80045d9c8/$FILE/LOM_fi.html). Viitattu 4.5.2003.
- [37] University of Princeton , WordNet. <http://www.cogsci.princeton.edu/~wn/>. Viitattu 15.1.2004
- [38] Valtiovarainministeriö, Hallinnon kehittämisosasto, Valtion tietotekniikan rajapintasuosituksia. Helsinki 2001. <http://www.vn.fi/vm/kehittaminen/tietohallinto/hko1.htm>. Viitattu 5.5.2003
- [39] Valtiovarainministeriö, Hallinnon kehittämisosasto, Nimeämiskäytäntö julkishallinnon sanastotyössä, <http://www.vm.fi/tiedostot/pdf/fi/34404.pdf>. Viitattu 6.8.2003
- [40] Valtiovarainministeriö, Hallinnon kehittämisosasto, Julkishallinnon XML-strategia. Työryhmämuistioita, 18/2003. Edita Prima Oy, Helsinki 2003.
- [41] W3C, Namespaces in XML 1.1. <http://www.w3.org/TR/xml-names11/>. Viitattu 9.9.2003
- [42] W3C, OWL Web Ontology Language Use Cases and Requirements, W3C Candidate. Recommendation 18 August 2003, <http://www.w3c.org/TR/webont-req/>, viitattu 8.12.2003
- [43] W3C, XML Schema Part 0: Primer. <http://www.w3.org/TR/xmlschema-0/>. Viitattu 9.9.2003
- [44] W3C, XML, <http://www.w3.org/TR/2000/REC-xml-20001006>. Viitattu 9.9.2003
- [45] W3C, World Wide Web Consortium julkaisee RDF- ja OWL-suositukset, <http://www.w3c.tut.fi/press/2004/0211-rdfowl/>. Viitattu 8.8.2004
- [46] Wrobel S., Concept Formation and Knowledge Revision. Kluwer Academic Publisher. Dordrecht 1994.

Liite 1.

Kuvan 25 mukaisen sisältökomponentin ominaispiirteiden sanalliset kuvaukset

CONCEPT NAME	: sisältökomponentti
FULL NAME	: sisältökomponentti
DESCRIPTION:	digitaalinen oppimateriaali tai sen osa
CONCEPT NAME	: nimike
FULL NAME	: nimike
DESCRIPTION	: komponentin sisältöä kuvaava nimi, voi sisältää yhden tai useampia sanoja
CONCEPT NAME	: Id
FULL NAME	: Id
DESCRIPTION:	komponentin yksilöivä tunnus HAMK:n järjestelmässä
CONCEPT NAME	: tek.oik.sopimus
FULL NAME	: tekijänoikeussopimus
DESCRIPTION:	tekijänoikeussopimuksen löytymiseen tarvittavat tiedot
CONCEPT NAME	: sop-tun
FULL NAME	: sopimuksen tunnus
DESCRIPTION:	
CONCEPT NAME	: sop-URL
FULL NAME	: sopimuksen absoluuttinen verkko-osoite
DESCRIPTION	: Tekijänoikeussopimuksen absoluuttiset verkko-osoitteet. Vähintään yksi verkko-osoite on ilmoitettava.
CONCEPT NAME	: URL
FULL NAME	: sisältökomponentin URL
DESCRIPTION	: Sisältökomponentin absoluuttiset verkko-osoitteet. Vähintään yksi verkko-osoite on ilmoitettava.
CONCEPT NAME	: julk.pvm
FULL NAME	: julkaisupäivämäärä
DESCRIPTION:	Päivämäärä, jolloin komponentti on talletettu järjestelmään.
CONCEPT NAME	: kuvaus
FULL NAME	: kuvaus
DESCRIPTION:	Komponentin sisällön ja tarkoituksen kuvaus.
CONCEPT NAME	: tekijä
FULL NAME	: komponentin tekijä
DESCRIPTION:	Komponenttien tekijän tiedot. Tekijöitä on vähintään yksi.
CONCEPT NAME	: HAMKista
FULL NAME	: HAMKista
DESCRIPTION	: Sisältää tiedon siitä, onko komponentin tekijä Hämeen ammattikorkeakoulun henkilökuntaan kuuluva vai ulkopuolinen. Tietoa tarvitaan haettaessa henkilön tietoja joko henkilöstöjärjestelmästä tai sidosryhmärekisteristä
CONCEPT NAME	: on
FULL NAME	: on

DESCRIPTION:

CONCEPT NAME : ei
FULL NAME : ei

DESCRIPTION:

CONCEPT NAME : henkilö
FULL NAME : henkilö
DESCRIPTION: Komponentin tekijän henkilötiedot.

CONCEPT NAME : sotu
FULL NAME : sotu
DESCRIPTION: Tekijän sosiaaliturvatunnus.

CONCEPT NAME : snimi
FULL NAME : sukunimi
DESCRIPTION: Tekijän sukunimi.

CONCEPT NAME : enimi
FULL NAME : etunimi
DESCRIPTION: Tekijän etunimi.

CONCEPT NAME : email
FULL NAME : email osoite
DESCRIPTION: Henkilön sähköpostiosoite.

CONCEPT NAME : puhelin
FULL NAME : puhelin
DESCRIPTION: Henkilön puhelinnumero.

CONCEPT NAME : ammatti
FULL NAME : ammatti
DESCRIPTION: tekijän ammatti

CONCEPT NAME : tutkinto
FULL NAME : tutkinto
DESCRIPTION: Korkein tiedossa oleva tekijän suorittama tutkinto.

CONCEPT NAME : tyyppi
FULL NAME : tyyppi
DESCRIPTION: Komponentin tarkoitusta kuvaava luonnehdinta: teoria, harjoitus, vastaus, koe.

CONCEPT NAME : avainsana
FULL NAME : avainsana
DESCRIPTION: Komponentin sisältöä kuvaavia sanoja. Vähintään yksi ilmoitettava.

CONCEPT NAME : dok.URL
FULL NAME : dok.URL
DESCRIPTION : Sisältökomponentin dokumentaation sisältävän tiedoston absoluuttiset verkko-osoitteet. Vähintään yksi osoite ilmoitettava, mikäli komponentti sisältää ohjelmakoodia tai on multimedialkooste.

CONCEPT NAME : luokitus
FULL NAME : luokitus
DESCRIPTION: oppiaine/aihekohtainen sisältöluokitus ylhäältä komponenttitasolle asti.

CONCEPT NAME : kompId
FULL NAME : sisältökomponentin tunnus
DESCRIPTION : Sisältökomponentin sisältämien muiden sisältökomponenttien tunnuksset.

Liite 2a.

LOM 1.0 /General-luokka, kansallinen sovellusprofiili

Nimi	kuvaus	Arvoavaruus	Tietotyyppi	Esimerkki
Identifier (Tunniste)	Globalisti yksikäsitteinen oppisisällön identifioiva tunnus			
Catalog (Luettelo)	Tunniste- tai luettelointijärjestelmän nimi tai tunnus. Nimeämisjärjestelmä	ISO/IEC 0646-1:2000:n valikoima	merkkijono, pienin sallittu maksimi:1000 merkkiä	"ISBN", "ADIADNE", "URI"
Entry (Merkintä)	Tunniste- tai luettelointijärjestelmän mukainen tunnisteiden arvo, joka viittaa tähän oppisisältöön tai yksilöi sen. Nimiavaruuteen kuuluva merkkijono.	ISO/IEC 0646-1:2000:n valikoima	merkkijono, pienin sallittu maksimi:1000 merkkiä	"2-7342-0318", "LEA0875", "http://www.ieee.org/documents/1234"
Language (Kieli)	Luonnollinen kieli tai kielet, joita on ensisijaisesti käytetty tässä oppisisällössä tarkoitetuille käyttäjille viestimiseen. Jos oppisisällön tarkoitus on muu kuin luonnollisella kielellä viestiminen, arvoksi annetaan "none" (ei mikään).	ISO 639:1988 tai RFC1766:1995	merkkijono, pienin sallittu maksimi:100 merkkiä	"en", "en-GB", "fi"
Description (Kuvaus)	Oppisisällön tekstimuotoinen kuvaus. HUOM: Kuvauksen tulisi olla kieleltään ja luonteeltaan sopiva niille, jotka päättävät, onko kuvailtava oppisisältö sopiva ja tarpeellinen käyttäjälle.		Kielimerkkijono, pienin sallittu maksimi:100 merkkiä	("fi", Tässä videoleikkeessä esitellään lyhyesti leonardo da Vincin elämä ja tuotanto. Pääpaino on hänen taiteellisilla tuotoksillaan, ennen muuta mona Lisalla"
Keyword (Avainsana)	Oppisisällön aihepiiriä kuvaava avainsana tai ilmaisu.		Kielimerkkijono, pienin sallittu maksimi:1000 merkkiä	("fi", "Mona Lisa")

Liite 2b.

LOM 1.0 /Lifecycle-luokka, kansallinen sovellusprofiili

LOM 1.0 /Lifecycle-luokka – kansallinen sovellusprofiiliNimi	kuvaus	Arvoavaruus	Tietotyyppi	Esimerkki
Status (Tila)	Oppisisällön valmistuneisuuden tila tai vaihe	draft(luonnos) final(lopullinen) revised(tarkastettu versio) unavailable(ei käytävissä)	sanasto	
Contribute (Osallistuja)	Ne tahot, jotka ovat vaikuttaneet tämän oppisisällön tilaan sen elinkaaren aikana			
Contribute.Role (Osallistujan rooli)	Vaikutuksen laji. HUOM: Ainakin oppisisällön tekijä(t) tulisi kuvata	author(tekijä) ... subject matter expert (sisältöasiantuntija)	sanasto	
Contribute.Entity(Osallistuja taho)	Tähän oppisisältöön vaikuttaneiden tahojen yksilöinti ja tietoja niistä. Tahojen tulee olla tärkeysjärjestyksessä siten, että merkittävin on ensimmäisenä.	vCard, sellaisena kun sen määrittelee IMC vCard 3.0 (RFC 2425, RFC 2426)	merkkijono, pienin sallittu maksimi:1000 merkkiä	"BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area Administrator\nAssistant\nEMAIL;TYPE=INTERNET:jfriday@host.com\nEND:VCARD\n"
Contribute.Date (Päiväys)	Vaikuttamisen päivämäärä		aika	"2001-08-23"

LOM 1.0 /MetaMetadata-luokka – kansallinen sovellusprofiili

Nimi	kuvaus	Arvoavaruus	Tietotyyppi	Esimerkki
Metadata Schema (Metatietorakennne)	Sen virallisen määrittelyn nimi ja versio, jota on käytetty tämän metatiedon luomiseen.	ISO/IEC 10646-1:2000: valikoima	Merkkijono (pienin sallittu maksimi:30 merkkiä)	"LOMv1.0"
Language (Kieli)	Metatietokuvauksessa käytetty kieli. EHTO: Kielen tulee olla sama kuin kuvauksen kielimerkkijonotyyppisten arvojen kieli.	ISO 639:1988 tai RFC1766:1995	Merkkijono (pienin sallittu maksimi:100 merkkiä)	"fi"

LOM 1.0 /Technical-luokka – kansallinen sovellusprofiili

Nimi	kuvaus	Arvoavaruus	Tietotyyppi	Esimerkki
Format (tallennusmuoto)	Tämän oppisisällön tekninen tietotyyppi tai tekniset tietotyypit. EHTO:Tätä tietoalkiota tulee käyttää määrittämään ohjelmisto, joka tarvitaan oppisisällön käsittelyyn.	RFC 2048:1996 tai "non-digital"	Merkkijono (pienin sallittu maksimi:100 merkkiä)	"video/mpg", "application/x-toolbook", "text/html"
Size (Koko)	Digitaalisen oppisisällön koko tavuina, joka viittaa oppisisällön todelliseen, ei pakattuun kokoon.	ISO/IEC 646:1991, ainoastaan numeroita "0".."9"	Merkkijono (pienin sallittu maksimi:30 merkkiä)	"5000"
Location (Sijainti)	Merkkijono, joka ilmaisee oppisisällön fyysisen sijainnin. Voi olla URL tai metodi, jonka perusteella voi ratkaista sijainnin.	ISO/IEC 10646-1:2000	Merkkijono (pienin sallittu maksimi:1000 merkkiä)	"http://host/id"

Liite 2c.

LOM 1.0 /Educational-luokka – kansallinen sovellusprofiili

Nimi	kuvaus	Arvoavaruus	Tietotyyppi	Esimerkki
Learning Resource Type (Oppimateriaalin laji)	Oppisisällön erityislaji tai lajit. Hallitsevin laji ensimmäisenä.	OED:1989 {exercise(harjoitus), ... lecture(luento)}	Sanasto	

LOM 1.0 /Rights-luokka – kansallinen sovellusprofiili

Nimi	kuvaus	Arvoavaruus	Tietotyyppi	Esimerkki
Cost (Maksut)	Vaatiiko oppisisällön käyttö maksun..	Yes (kyllä) no (ei)	Sanasto	
Copyright and Other Restrictions (Tekijän-oikeudet ja muut rajoitukset)	Koskevatko tekijänoikeudet ja muut rajoitukset tämän oppisisällön käyttöä.	Yes (kyllä) no (ei)	Sanasto	
Description (Kuvaus)	Oppisisällön käyttöehtoihin liittyviä kommentteja.		Kielimerkki-jono (pöytäkirja sallittu maksimi:1000 merkkiä)	("fi", "Tämän oppisisällön käyttö on sallittua vain Amnesty International – järjestölle tehdyn lahjoituksen jälkeen)

LOM 1.0 /Relation-luokka – kansallinen sovellusprofiili

Nimi	kuvaus	Koko	Arvoavaruus	Tietotyyppi	Esimerkki
Kind (Luonne)	Tämän oppisisällön ja kohdassa 7.2:relation.Resource yksilöidyn oppisisällön välisen suhteen luonne.	1	Perustuu DC: ss partof:on...:n osa isrequired by: on...:n vaatima	järjestyksetön sanasto	
Resource (Kohde)	Se kohteena oleva oppisisältö, johon tämä suhde viittaa	1			
Resource Identifier (Tunniste)	Gloobaalisti yksikäsitteinen tunnus, joka yksilöi kohteena olevan oppisisällön	psm: 10 alkioita			
Identifier Catalog (Luettelo)	Tässä kohdassa käytetyn tunniste- tai luettelointijärjestelmän nimi tai tunnus. Nimeämisjärjestelmä.	1	ISO/IEC 10646-1: 2000:n valikoima	merkkijono (psm: 1000 merkkiä)	"ISBN", "ARIADNE", "URI"
Identifier Entry (Merkintä)	Tunniste- tai luettelointijärjestelmän mukainen tunniste arvo, joka viittaa kohteena olevaan oppisisältöön tai yksilöi sen. Nimiavaruuteen kuuluva merkkijono.	1	ISO/IEC 10646-1: 2000:n valikoima	merkkijono (psm: 1000 merkkiä)	"2-7342-0318", "LEA0875", "http://www.ieee.org"
Resource Description	Kohteena olevan oppisisällön kuvaus	10		Kielimerkkijono (psm:1000 merkkiä)	("fi", "QuickTime-elokuva Mona-Lisasta ..)

Liite 2d.

LOM 1.0 /Classification-luokka – kansallinen sovellusprofiili

Nimi	kuvaus	Arvoavaruus	Tietotyyppi	Esimerkki
Purpose (tarkoitus)	Oppisisällön luokittamisen tarkoitus.	discipline (oppiaine) idea (idea) prerequisite(ennakkovaatimuk.)ed ucational objektive (opetuksellinen tavoite) accessibility (saavutettavuus) restrictions (rajoitukset) educational level (koulutusaste) skill level (taitotaso) security level (suojaustaso) competency (kompetenssi)	Sanasto	
Taxon Path (Luokituspolku)	Luokituspolku jossakin erityisessä luokitusjärjestelmässä. kukin taso on edeltävällä tasolla olevan määritelmän tarkennus.			
TaxonPath. Source (Luokituspolun lähde)	Luokitusjärjestelmän nimi. Tämä tietoalkio voi käyttää mitä tahansa tunnistettua "virallista" luokitusta tai käyttäjän määrittelemää luokitusta.	ISO/IEC 10646-1:2000	Kielimj. psm: 1000 merkkiä)	("en","ACM"), ("en","MESH"), ("en","ARIADNE")
TaxonPath. Taxon (Luokituspolun luokitusmerkintä)	Erityinen luokitukseen sisältyvä termi. Luokitusmerkintä on solmu, jolla on määritelty nimiö tai termi. Luokitusmerkintöjen järjestetty luettelo luo taksonomisen polun, joka johtaa luokittelussa yleisemmästä merkinnästä yksityiskohtaisempaan merkintään			{["12",["fi","Fysiikka"]], ["23",["fi","Akustiikka"]], ["34",["fi","Laitteet"]], ["45",["fi","Instrumentti"]]}.
Taxon.Id (Luokitusmerkinnän identifiointi -tunnus)	Luokitusmerkinnän tunnus, kuten luku tai kirjainyhdistelmä, joka saadaan luokituksen lähteestä.	ISO/IEC 10646-1:2000	Merkkijono (psm: 100 merkkiä)	"320", "4.3.2", "BF180"
Taxon.Entry (Luokitusmerkinnän tietue)	Luokitusmerkinnän nimiö tekstinä		Kielimj. (psm: 500 merkkiä)	("fi","Lääketiede")