

Lääkitystiedon tietoarkkitehtuurin suunnittelu

Joonas Mäkinen

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tietojärjestelmätiede
Pro gradu -tutkielma
Ohjaaja: Pirkko Nykänen
Syyskuu 2009

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Tietojärjestelmätiede

Joonas Mäkinen: Lääkitystiedon tietoarkkitehtuurin suunnittelu

Pro gradu -tutkielma, 67 sivua, 8 liitesivua

Syyskuu 2009

Lääkitystieto on tärkeä hoitoprosessin osa, jolla on yhteys hoidon onnistumiseen ja potilasturvallisuuteen. Terveydenhuollon ammattilaisella tulisi olla käytössään ajantasainen, kattava ja luotettava tieto potilaan kokonaislääkityksestä. Tämän tavoitetilan saavuttaminen edellyttää kohdealueen ymmärtämystä, semanttisesti yhdenmukaista käsitteistöä ja yhteistä tietomallia eri tietojärjestelmien organisaatioiden välillä. Tämän tutkimuksen aiheena on lääkitystiedon tietoarkkitehtuuri, jonka on tarkoitus omalta osaltaan tukea lääkitystiedon hallinnan kehittämistä. Tietoarkkitehtuuri on tuotettu terveydenhuollon ammattilaisen näkökulmasta.

Tutkimus on tehty suunnittelutieteellisenä tutkimuksena, jossa valittuja teorioita soveltamalla muodostetaan lääkitystiedon tietoarkkitehtuuri. Aluksi tutkimuksessa tutustutaan lyhyesti terveydenhuollon tietojärjestelmien ja tiedonhallinnan nykytilaan kansallisesti ja kansainvälisesti erityisesti lääkitystiedon hallinnan näkökulmasta. Seuraavaksi tutkitaan lääkitystiedon käsitettä, lääkitystiedon käyttöä ja tietotarpeita. Lopuksi suunnitellaan valittuihin malleihin perustuen lääkitystiedon tietoarkkitehtuuri.

Tutkimus perustuu kirjalliseen aineistoon ja Tekesin FinnWell-rahoitteisessa Lääkitystiedon hallinta -hankkeessa tehtyyn työhön. Hankkeessa Tampereen yliopisto kehitti lääkitystiedon prosessi- ja tietomallit. Nämä mallit ovat tässä pro gradu -tutkielmassa toteutetun lääkitystiedon tietoarkkitehtuurin perustana.

Avainsanat ja -sanonnat: Tietoarkkitehtuuri, terveydenhuollon tietojärjestelmät, lääkitystieto, kokonaislääkitys, HL7 RIM.

Sisällysluettelo

1	Johdanto.....	1
1.1	Terveydenhuollon tietojärjestelmät ja tiedonhallinta Suomessa – nykytila.....	1
1.2	Lääkitystiedon hallinta – kansainvälinen katsaus.....	3
1.3	Lääkitystiedon hallinta -tutkimushanke.....	5
2	Lääkitystieto	7
2.1	Lääkitystiedon käsitteestä.....	7
2.2	Lääkitystieto terveydenhuollon ammattilaisen näkökulmasta.....	8
2.3	Lääkitystiedon hallintamallit	10
2.4	Lääkitystiedon lähteet.....	12
2.5	EU:n suositukset potilaan lääkitystiedoiksi.....	13
3	Tutkimusongelma, rajaukset ja tutkimusmenetelmät.....	17
3.1	Tutkimusongelma ja sen rajaus	17
3.2	Tutkimusmenetelmät	17
3.3	Mallinnusmenetelmät	18
4	Tutkimuksen taustateoriat, mallit ja viitekehykset.....	21
4.1	Kokonaisarkkitehtuuri	21
4.2	Arkkitehtuuriviitekehys	23
4.2.1	Zachmanin viitekehys	23
4.2.2	RM-ODP.....	25
4.2.3	Generic Component Model.....	26
4.3	Tietoarkkitehtuuri	27
4.4	Arkkitehtuurin suunnittelumenetelmiä.....	29
4.4.1	HIS-DF.....	32
4.4.2	ISO/TC 215.....	33
4.4.3	HL7 Development Framework (HDF)	34
4.5	Terveydenhuollon tietoarkkitehtuuriviitemallit.....	36
4.6	Tietoarkkitehtuuri tiedon esittämisen ja käytön näkökulmasta.....	38
4.7	Tietoarkkitehtuurin kehittämisprosessi	40

5	Lääkitystiedon tietoarkkitehtuuri	44
5.1	Kohdealueen tutkimus ja rajaaminen	44
5.2	Lääkitystiedon prosessimalli	45
5.3	Lääkitystiedon tietomalli	45
5.4	Lääkitystiedon harmonisointi viitemalleihin	46
5.5	Lääkitystiedon tietoarkkitehtuuri.....	51
6	Yhteenveto ja johtopäätökset	55
	Viiteluettelo	57
	Liite 1: HL7 RIM versio 2.26.....	65
	Liite 2: Lääkitystiedon prosessimallit.....	66
	Liite 3: Lääkitystiedon tietomalli	70
	Liite 4: Medication D-MIM v1.4.7.....	72

1 Johdanto

Lääkitys on keskeinen osa hoitoprosessia. Oikealla lääkityksellä on yhteys hoidon onnistumiseen sekä potilasturvallisuuteen, ja siksi terveydenhuollon ammattilaisella tulisi olla saatavilla kattava ja luotettava tieto potilaan kokonaislääkityksestä. Ajantasainen lääkitystieto koostuu useasta osatekijästä, joiden perusteella lääkäri muodostaa käsityksen kokonaislääkityksestä. Varmistuaakseen lääkityksen oikeellisuudesta lääkäri tai hoitaja saattaa joutua hakemaan tietoa monesta eri lähteestä. Kaikkea tarpeellista tietoa, kuten toisessa terveydenhuollon yksikössä määrättyjä lääkkeitä, ei ole aina saatavissa tai tiedon hankkimiseen menee aikaa [Aarnio, 2008a]. Kehittämällä tiedonkulkua ja yhdenmukaistamalla lääkitystietojen kirjaamista, tietosisältöä ja käyttöä voidaan parantaa lääkitystiedon hallintaa.

Lääkitystietojen kattavuus, yhdenmukaisuus ja välittäminen organisaatioiden välillä edellyttävät kohdealueen ymmärtämistä, semanttisesti yhdenmukaista käsitteistöä ja yhteistä tietomallia. Tämän tutkimuksen tuloksena on tätä tarkoitusta palveleva lääkitystiedon tietoarkkitehtuuri. Aluksi tutkitaan tietoarkkitehtuurille perustan rakentavia osaluueita, kuten kansalliset suuntaviivat terveydenhuollon tietojärjestelmille sekä nykytila, lääkitystiedon käsite, lääkitystiedon tietotarpeet ja käyttö. Sen jälkeen tutustutaan tietoarkkitehtuurille teoriapohjan muodostaviin arkkitehtuuriviitekehyksiin, viitemalleihin ja mallinnusmenetelmiin, joihin tukeutuen muodostuu käsitys tietoarkkitehtuurin syntyprosessista.

1.1 Terveydenhuollon tietojärjestelmät ja tiedonhallinta Suomessa – nykytila

Suomen terveydenhuollossa informaatio- ja kommunikaatioteknologioiden käyttö on kasvanut voimakkaasti viime vuosina ja käytössä on laajasti erilaisia sähköisiä järjestelmiä. Vuodelta 2007 kerättyjen tietojen mukaan [Winblad ja muut, 2008] mukaan sähköinen potilaskertomusjärjestelmä on kattavasti käytössä sairaanhoitopiireissä, perusterveydenhuollossa ja myös kaikilla kyselyyn vastanneilla yksityisillä toimijoilla. Potilaskertomusjärjestelmiä myös käytetään aktiivisesti ja vaikuttaisi siltä, että valtaosa potilaiden tiedoista kirjataan nykyisin sähköisessä muodossa. Se tarkoittaa myös, että tietoa potilaan aiemmin saamasta hoidosta löytyy sähköisenä yhä enemmän. Potilaskertomusjärjestelmiä on käytössä useita, esimerkiksi Effica, Mediatri, Miranda ja Pegasos.

Raportin mukaan [Winblad ja muut, 2008] 17/21 sairaanhoitopiireistä on käytössä jokin aluetietojärjestelmä. Aluetietojärjestelmässä voi muun muassa olla toteutettuna katseluyhteys toisen organisaation potilaskertomusteksteihin aluetietojärjestelmään liittyneiden terveydenhuollon toimijoiden välillä, mutta potilaskertomustietoja välitetään muillakin tavoin. Laajalti käytössä on myös lähete-palautejärjestelmiä, joissa lähetteen

antanut organisaatio saa potilastietojärjestelmään palautteen hoidon antaneesta yksiköstä, esimerkiksi terveyskeskus erikoissairaanhoidosta. Lähetteen ja palautteen mukana voi kulkea tietoa annetusta hoidosta ja lääkityksestä. Toimijoiden välillä välitetään aktiivisesti myös kuvantamistietoja ja laboratoriotuloksia.

Alueellisten tietojärjestelmäratkaisujen rinnalle on tullut kansallinen tietojärjestelmäarkkitehtuuri, jonka rakentamista ohjaavat laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä [Laki 159/2007] sekä laki sähköisestä lääkemääräyksestä [Laki 61/2007]. Kyseessä on palvelukeskeinen arkkitehtuuri, jossa itsenäisiä osajärjestelmiä tullaan liittämään toisiinsa. Potilasasiakirjat tullaan tallentamaan keskitetysti Kansallinen terveysarkisto KanTaan. Sen kautta yhden organisaation potilasasiakirjoja voidaan potilaan suostumuksella hakea katseltavaksi myös muissa organisaatioissa. Tarkoitusta varten on kehitetty mm. avoimet CDA R2 -rajapinta-, tietoturva- ja ydintietomäärittelyt. Kannan muita palveluita ovat viestinvälitys, tunnistaminen ja sähköinen allekirjoitus, hakemisto ja rekisteröinti, suostumustenhallinta, loki ja valvonta sekä koodistopalvelu. [STM, 2007a] Arkkitehtuurihanketta toteutetaan STM:n ohjauksessa. Kansallisena toimijana on Kela, joka vastaa sähköisen arkiston ja reseptikeskuksen kehittämisestä.

Osana kansallista tietojärjestelmäarkkitehtuuria otetaan käyttöön sähköinen resepti. Potilas saa jatkossa mukaansa reseptin sijaan potilasohjeen. Resepti tallentuu Kelan hallinnoimaan keskitettyyn reseptitietokantaan, josta apteekki noutaa sen asiakkaan tullessa noutamaan lääkettä. Hyötyinä katsotaan olevan kokonaislääkitystietoisuuden parantuminen ja lääkitysvirheiden riskin vähentyminen siksi, että terveydenhuollon organisaatioissa ja apteekissa voidaan paremmin huomata päällekkäiset lääkitykset. Lääkärillä on tulevaisuudessa mahdollisuus tarkastella reseptikeskuksen tietoja potilaskertomusjärjestelmän kautta, mistä hän voi selvittää esimerkiksi reseptin toimitustiedot. Katseluyhteys omiin resepteihin on suunnitteilla myös kansalaiselle. [STM, 2008; HE, 2006] Tarvittaessa paperi- ja puhelinreseptejä voidaan tehdä myös jatkossa.

Sähköinen resepti ei ratkaise kaikkia lääkitystiedon hallinnan haasteita, sillä nimensä mukaisesti reseptikeskuksessa on reseptejä. Siellä ei ole tietoa sairaalalääkkeistä tai käsikauppalääkkeistä eikä reseptiin tarvitse kirjata indikaatiota eli lääkkeen käyttöaihetta. Reseptiin kirjattu tieto lääkkeen annostuksesta ei pidä sekään välttämättä paikkaansa, koska muutoksia annostukseen voidaan tehdä kirjoittamatta uutta reseptiä. Lääke ei välttämättä ole käytössä, vaikka resepti olisi voimassa. Reseptien perusteella ei voi luotettavasti muodostaa kuvaa potilaan lääkityshistoriasta tai kokonaislääkityksestä, koska resepteistä voi puuttua lääkitykseen liittyvää hyvin tärkeitäkin tietoa. [Nykänen ja muut, 2008; STM, 2008]

Potilaskertomustiedon kirjaamistavat ja tietosisältö ovat olleet epäyhtenäiset eri organisaatioiden käyttämissä potilastietojärjestelmissä ja tietoa on kirjattu paljon rakenteettomasti. Rakenteisella ydintiedolla ja kirjaamiskäytännöllä parannetaan tiedon käyttömahdollisuuksia. Kertakirjaamisperiaatteen seurauksena vähenee kirjaamistyön tarve,

koska kerran kirjattuja tietoja voidaan täyttää valmiiksi lomakepohjiin ja vähentää turhaa päällekkäistä kirjaamista muissakin tilanteissa. Rakenteisen tiedon käyttöpoteentiaali on suuri, sen avulla voidaan toteuttaa päätöksenteon tukijärjestelmiä ja varoituksia liittyen potilaan lääkeaineallergioihin. Kertakirjaamisen ja päätöksenteon tuen yhdistelmä voi vähentää myös kirjaamisvaiheessa syntyneiden virheiden määrää. Rakenteiseen kirjaamiseen siirtyminen edellyttää myös yhteisiä koodistoja ja luokituksia, joiden jakelua varten on avattu kansallinen koodistopalvelu. [STM, 2006; STM, 2007b]

1.2 Lääkitystiedon hallinta – kansainvälinen katsaus

Eri maiden terveydenhuollon kansalliset tietoarkkitehtuurimallit voi jakaa karkeasti kolmeen ryhmään tiedon hallintatavan mukaan: a) keskitetty malli, jossa on valtakunnallisesti keskitetty potilaskertomusjärjestelmä tai sähköinen arkisto; b) hajautettu malli, jossa tieto säilytetään perusjärjestelmissä ja tiedonsaanti on järjestetty linkkien tai viitteiden avulla; sekä c) korttimalli, jossa potilasta koskevat keskeiset tiedot sekä viitteet muihin tietoihin tallennetaan potilaan hallitsemalle sirukortille tai muulle välineelle [STM, 2006]. Lyhyt muutamien EU-maiden lääkitystietoon liittyviä suunnitelmia koskeva tarkastelu tukee tätä jaottelua, joskaan järjestelmiä ei voi kokonaan kategorisoida yhden mallin alle. Suomen nykyinen ratkaisumalli, KanTa-arkkitehtuuri, vastaa näistä kolmesta lähinnä keskitettyä, toisaalta tieto säilyy alkuperäisessä järjestelmässä sen lisäksi että keskeiset tiedot tallennetaan kansalliseen arkistoon. KanTa-arkkitehtuuri onkin eräänlainen hybridimalli johtuen siitä, että se on riippuvainen perusjärjestelmistä ja sen arkkitehtuuri on palvelukeskeinen [Mäki, 2008]. Arkkitehtuuria voisi pitää keskitetynä vasta, jos käytössä olisi yhteinen potilaskertomusjärjestelmä tai jos perusjärjestelmät käyttäisivät yhteistä arkistoa ensisijaisena tiedonlähteenä ja säilytyspaikkana eikä sinne toimitettaisi vain sovittuja osia potilasasiakirjoista, kuten Kannassa on suunniteltu tehtävän. Saksassa ja Sloveniassa on käytössä terveyskorttipohjainen järjestelmä [eHealth ERA, 2007] ja Hollannissa hajautettu malli, jossa tiedot säilytetään perusjärjestelmissä ja saadaan muualla nähtäväksi keskitetyn palvelun viitteiden (switch point) avulla [Jong, 2007]. Terveyskortti on käytännössä yhdistelmä kortilla olevia perustietoja ja keskitettyjä tai hajautettuja tietokantoja, joiden sisältämät tiedot voi löytää kortin sisältämien tietojen avulla. Vaikka tässäkään mallissa suuri osa tiedoista ei sijaitse kortilla, sen tulkitseminen omaksi mallikseen on perusteltua, koska kortilla on keskeinen rooli tiedon käytössä ja hallinnassa. Kansalainen kontrolloi suostumustansa tietojensa luovuttamiseen päättämällä kortin käytöstä.

Euroopan komission eHealth ERA -hankkeen raportissa [2007] käsitellyistä maista lähes kaikilla lääkityksen hallinta on jollakin laajuudella mukana suunnitelmissa. Yhteistä monille EU-maiden kansallisille ratkaisuille vaikuttaisi olevan se, että lääkitystiedon hallinnasta on keskeisesti nostettu esiin lähinnä sähköinen resepti ja sen sisältämän tiedon eriateinen hyödyntäminen potilaan lääkityksen hallinnassa. Ongelmat lääkitys-

tiedon hallinnan kanssa on siis tiedostettu, mutta reseptienhallintaa kokonaisvaltaisemmista ratkaisuksista ei löydy mainintoja. [eHealth ERA, 2007; Sjöborg et al., 2007; Sundhed.dk, 2008; Sušelj, 2006]

Ruotsissa terveydenhuollosta vastaavat läänit. Kansallinen hanke on nimeltään Carelink, joka yhdistää alueiden lisäksi myös terveydenhuoltoa ja kuntien vastuulla olevan vanhusten ja vammaisten hoidon palveluita. Tieto kulkee Sjunet-verkossa, joka on IP-pohjainen internetistä erillinen verkko, joka tarjoaa muun muassa autentikointipalvelut. Verkkoinfrastruktuurin perustalle on toteutettu sähköinen resepti, radiologian etäpalvelut ja videoneuvottelujärjestelmä [Larson et al., 2003]. Ruotsissa on kehitetty sähköisen reseptin järjestelmää jo useita vuosia ja yli 70 % lääkemääräyksistä toimitettiin sähköisesti vuoden 2005 lopussa [Sjöborg et al., 2007]. Lääkkeen määräystä ohjaa ja avustaa lääkärin näkemä tieto potilaalle eri organisaatioissa määrätystä lääkkeestä, päätöksenteon tuki ja potilaan muut tiedot, kuten pituus, paino, allergiat ja yliherkkyydet. Lääkemääräys toimitetaan sähköiseen ”postilaatikkoon”, josta se noudetaan apteekissa. Potilaalla ei ole katseluyhteyttä resepteihinsä. Ruotsin sähköisen reseptin käyttöönottoa on helpottanut huomattavasti se, että apteekit ovat valtion monopoli ja järjestelmän kehittäminen on siten ollut yksinkertaisempaa kuin esimerkiksi Suomessa, jossa apteekit ovat yksityisiä yrityksiä ja apteekkijärjestelmiä on useita [HE, 2006].

Tanska aloitti sähköisen reseptin käyttöönoton 1995–1996 osana kansallista Med-Com-hanketta ja maa on ollut edelläkävijöitä sen käytössä. Neljä viidesosaa resepteistä toimitetaan sähköisesti suoraan apteekkiin [sundhed.dk, 2008]. Hyppösen ja muiden [2005] mukaan myös Tanskassa on suunnitteilla Suomen tapaan keskitetty reseptipalvelin. Nykyisen järjestelmän uudistaminen on aloitettu, sillä se alkaa olla ikääntynyt ja siinä on joitakin puutteita, kuten ettei reseptin toimitustietoja ole saatavilla, apteekkia ei voi valita vapaasti ja myös uusimiskäytännössä on parannettavaa [HE, 2006]. Uusi Health Data Network yhdistää terveydenhuollon toimijat. Kansalaisilla on mahdollisuus päästä käsiksi joihinkin tietoihinsa sundhed.dk-portaalin kautta, esimerkiksi hoitohistoriaan ja henkilökohtaiseen lääkitysprofiiliin. Ammatillaiset voivat palvelun kautta katsoa muun muassa potilaan lääkitystietoja ja laboratoriotuloksia. Järjestelmän kautta on ammattilaisilla myös pääsy tarkastelemaan paikallisten potilastietojärjestelmien tietoja potilaista. [sundhed.dk, 2008]

Sloveniassa on käytössä HIC-kortti (Health Insurance Card), joka jaetaan kaikkiaan noin kahdelle miljoonalle asukkaalle. Kortille tallennetaan tietoa potilaan hoidosta, tieto määrätystä lääkkeestä, allergioista, yliherkkyyksistä ja elinluovutuksen sallimisesta. Terveydenhuollon ammattilaisilla on oma korttinsa. Tieto määrätystä lääkkeestä – käytännössä resepteistä – kirjautuu sekä kortille että kansalliseen tietokantaan. Käytössä on itsepalvelupäätteitä, joiden avulla kortin lääkemääräykset voi päivittää keskitetyn tietokannan mukaisiksi. Kortille kirjataan kaikki määrättyt lääkkeet HIV-lääkitystä lukuun ottamatta. Enintään merkintöjä voi olla 46 kappaletta. Merkintään kuuluu lääkkeen tun-

niste, määrä, määräyspäivämäärä, lääkkeen määrääjän tunniste ja lääkkeen luovuttaneen apteekin tunniste. [Sušelj, 2006]

Saksassa on käytössä sähköinen terveystietokortti (eGesundheitskarte, eGK), joka on tarkoitus jakaa kaikille maan asukkaille. Terveystietokortin ammattilaiset saavat ammattilaiskortin, jolla voi tehdä sähköisen allekirjoituksen. Ensimmäisten ominaisuuksien joukossa käyttöön otetaan sähköinen resepti, joka tallentuu joko suoraan kortille tai viitteenä. Riippumatta siitä, kumpi tieto kortille tallennetaan, kortti on avain tietoihin pääsulle. Korttimallin etu voi olla, että se on kansalaiselle yksinkertainen. Esimerkiksi reseptin kuljettaminen entisen paperisen sijasta kortilla lääkäristä apteekkiin on helppo omaksua. Jatkossa kortille voidaan kirjata apteekissa sieltä ostetut käsikauppalääkkeet. Kortille tullaan myös tallentamaan akuutihoidossa tarvittavaa tietoa lääkeyliherkkyyksistä, allergioista ja kroonisista sairauksista. Sähköisen reseptin käytöstä tulee pakollista. [eHealth ERA, 2007; Hyppönen ja muut, 2006; HE, 2006]

Hollannissa terveydenhuollon kansallinen tietojärjestelmäratkaisu tulee perustumaan kansalliseen sähköiseen potilaskertomusjärjestelmään, joka antaa terveydenhuollon ammattilaisten tarkastella tietoja paikasta riippumattomasti. Potilaiden tiedot säilytetään perusjärjestelmissä ja linkitetään kansallisen tason ratkaisulla. NICTIZ (Dutch National Institute for Healthcare) kokoaa hajallaan olevat hankkeet yhteen tavoitteenaan maanlaajuinen potilastiedon vaihto perustuen kansalliseen sähköiseen potilaskertomukseen (EPR). Hanke jakautuu kahteen osaan: AORTAan, joka vastaa IT-infrastruktuurin suunnittelusta, ja terveydenhuollon sovelluksiin. NICTIZ aloitti kansallisten terveydenhuollon sovellusten kehittämisen fokuoimalla lääkitystietoon tarkoituksena suunnitella sähköinen lääkitystietoarkisto EMD (Electronic Medication Record tai eMedication Record). Sen avulla reseptitietoja voidaan vaihtaa yleislääkärin, sairaalan ja apteekkien välillä. Lääkitystietoarkistoon liitetään myös EPS, sähköisten lääkemääräysten tekoon tarkoitettu järjestelmä, jonka avulla lääkäri voi muun muassa katsoa, mitä reseptilääkkeitä potilas on hakenut apteekista. [NICTIZ, 2002; NICTIZ, 2005; Jong, 2007]

1.3 Lääkitystiedon hallinta -tutkimushanke

Tutkimukseni liittyy Tekes FinnWell -rahoitteiseen Lääkitystiedon hallinta -hankkeeseen, joka on ollut käynnissä vuosina 2006–2008 (Tutkimusrahoituspäätökset 40452/05, 40345/06). Hankkeen tarkoituksena on ollut ”täydentää kansallisia lääkitystiedon hallintaa tukevia hankkeita ... sovellus- ja prosessitason ratkaisuille, jotka tukevat kansallisissa hankkeissa kehitettyihin infrastruktuuriratkaisuihin. Keskeinen tutkimuskohde on potilaskohtainen lääkitystieto, jonka tallennusta, välitystä ja käyttöä pyritään tehostamaan ja järjeistämään eri terveydenhuollon prosesseissa ja eri terveydenhuollon toimijoiden näkökulmasta. Keskeisessä osassa on myös potilas, joka on lääkitystiedon keskeinen subjekti.” [Suomi ja muut, 2006] Hankkeen vetovastuussa toimi Turun kauppakorkeakoulu, jonka lisäksi mukana olivat Turun yliopisto, Oulun yliopis-

to, Tampereen yliopisto sekä Fujitsu Services Oy, Itella Suomi Oy, PharmaService Oy, Suomen Apteekkariliitto, TeliaSonera Oyj, Turun kaupungin terveystoimi, Turku Science Park ja Varsinais-Suomen sairaanhoitopiiri. Tutkimus jakautui viiteen osahankkeeseen, joista neljäs, Lääkitystiedon tietoarkkitehtuuri oli Tampereen yliopiston vastuulla ja jossa osahankkeessa myös tämän tutkimuksen perustana olevat prosessi- ja tietomallit on kehitetty. Muissa osahankkeissa selvitettiin muun muassa kansalaisen näkökulmaa (Turun yliopisto), lääkitystiedon hallintamalleja (Turun kauppakorkeakoulu), lääkitystiedon mobiilikäyttöä ja -sovelluksia (Oulun yliopisto) sekä koneellista annosjakelua (Turun kauppakorkeakoulu). Osahankkeiden välillä on tehty yhteistyötä ja vaihdettu tietoa. Hankkeen tulokset on kerätty kokoomateokseen *Lääkitys kohdalleen – Tietojen hallinta avain oikeaan lääkitykseen* [Suomi ja Raitoharju, 2008]. Tampereen yliopiston osahankkeen keskeiset tulokset löytyvät kirjan artikkelista Lääkitystiedon tietomalli [Nykänen ja muut, 2008].

Tässä pro gradu -tutkielmassa kehitetty tietoarkkitehtuuri perustuu osahankkeen aiemmissa vaiheissa tehtyihin prosessi- ja tietomalleihin. Tampereen yliopiston osahanke on edennyt perinteisen systeemityön mallin mukaisesti ja se vaiheistettiin pääpiirteissään seuraavasti:

1. 2006 Taustatutkimus: Lääkitystiedon integrointi potilastietojärjestelmiin.
 - Lääkitystiedon käsittely nykyisissä potilastietojärjestelmissä. Lääkitystiedon käsittelyn tavoitetilan määrittely huomioiden tiedon sisältö ja käyttö hoitoprosesseissa.
 - Yksityiskohtaiset tulokset on esitelty hankkeen sisäisessä raportissa [Nykänen ja Mäkinen, 2007].
2. 2007 Prosessimallinnus ja lainsäädännölliset ym. vaatimukset:
 - Lääkitystiedon prosessimallit terveydenhuollon ammattilaisen, erityisesti lääkärin, näkökulmasta. Tallennetun ja käytetyn tiedon tunnistaminen prosesseissa. Lainsäädännön, toiminnallisuuden ja uusien määrittelyiden asettamat vaatimukset lääkitystiedon sisällölle ja käsittelylle.
 - Yksityiskohtaiset tulokset on esitelty hankkeen sisäisessä raportissa [Mäkinen ja Nykänen, 2008] ja kansainvälisessä konferenssiesityksessä [Mäkinen and Nykänen, 2009].
3. 2008 Lääkitystiedon tietomalli ja tietoarkkitehtuuri:
 - Lääkitystietojen analysointi ja yhdenmukaistaminen perustuen ydin-tieto, HL7 ym. määrittelyyn. Tietomalli.
 - Tietomallia on käsitelty yksityiskohtaisesti hankkeen sisäisessä raportissa [Mäkinen ja muut, 2008].
 - Tietoarkkitehtuurin osalta tutkimus on tehty hankkeen päättymisen jälkeen ja tutkimus ja sen tulokset julkaistaan tänä pro gradu -tutkielmana.

2 Lääkitystieto

2.1 Lääkitystiedon käsitteestä

Lääkitystiedolla viitataan tietyn potilaan lääkitykseen. Lääkitystieto käsittää potilaan voimassaolevan lääkityksen sekä lääkityshistorian. Aarnio [2007a] määrittelee lääkitystiedon ”potilaskohtaiseksi tietämykseksi lääkkeen geneerisestä ja kaupallisesta nimestä, annostuksesta ja ottamistavasta sekä lääkemääräyksen hoidollisesta kontekstista”. Hoidollinen konteksti tarkoittaa ”lääkityksen aihetta, hoidon tavoitetta, määräämisympäristöä (kuka, millä ammattitaidolla, missä palvelussa) ja lääkityspäätökseen johtaneita olennaisimpia johtopäätöksiä”. Geneerinen nimi tarkoittaa lääkkeen vaikuttavaa ainetta. Niitä voi olla samassa lääkkeessä useampia.

Lääkitystieto-käsitettä käytetään melko yleisesti virallisissa yhteyksissä, mutta varsinaista määrittelyä sille ei näistä löydy. Sen voi tulkita siten, että käsite on siinä määrin vakiintunut asiantuntijoiden keskuudessa, että määrittelyä ei koeta tarpeelliseksi [Aarnio, 2007a]. Toinen yleisesti esiintyvä käsite on kokonaislääkitys, jolla on hieman eri merkitys kuin lääkitystiedolla. Lääkitystieto viittaa yleisesti potilaan lääkitystietoon siten kuin edellä on kuvattu, mutta kokonaislääkityksellä tarkoitetaan eräänlaista kattavaa yleiskuvaa potilaan lääkityksestä, lääkitystiedon tietojoukkoa, jonka lääkäri muodostaa ymmärtääkseen potilaan ajankohtaisen tilanteen ja lääkityshistorian tarpeellisin osin.

Tämän tutkimuksen kannalta englanninkielen vastine ”medication information” on osuvin siksi, että se ei sotke lääkkeen ja lääkityksen käsitettä toisiinsa. Lääkitystieto on tavallaan yläkäsite lääketiedolle. Silti sellaistaakin lääketietoa voi olla, jolla ei ole merkitystä lääkitystiedon kannalta.

Ydintietomäärittelyssä [STM, 2007b] käytetään käsitettä ”lääkehoito”. Lääkehoidon ydintiedot sisältävät lääkkeen nimen, vahvuuden, ATC-koodin, roolin, annostuksen, aloitus- ja muutospäivämäärän, aloituksen tai muutoksen syyn (ICD-10 indikaatio), lopetuspäivämäärän, vaihdettavuuden, lääkitysohjeistuksen antamisen ja lääkkeen määräjän ja organisaation [STM, 2007c].

Lääkitystiedon ydintiedoista voidaan koota lääkityslista [STM, 2007b]. Lääkkeet jakautuvat silloin neljään ryhmään (Kuva 1). Lääkitystiedon prosessi- ja tietomallinnusvaiheessa [Nykänen ja muut, 2008; Mäkinen and Nykänen, 2009] on asiantuntijapalautteen perusteella vaihdettu käsite pysyväislääkitys pitkäaikaislääkitykseksi, jota myös seuraavassa käytetään:

- 1) Pitkäaikaislääkitys (pysyväislääkitys)
- 2) Pitkäaikaislääkitys (tarvittaessa otettava)
- 3) Tilapäislääkitys (parhailaan käytössä)

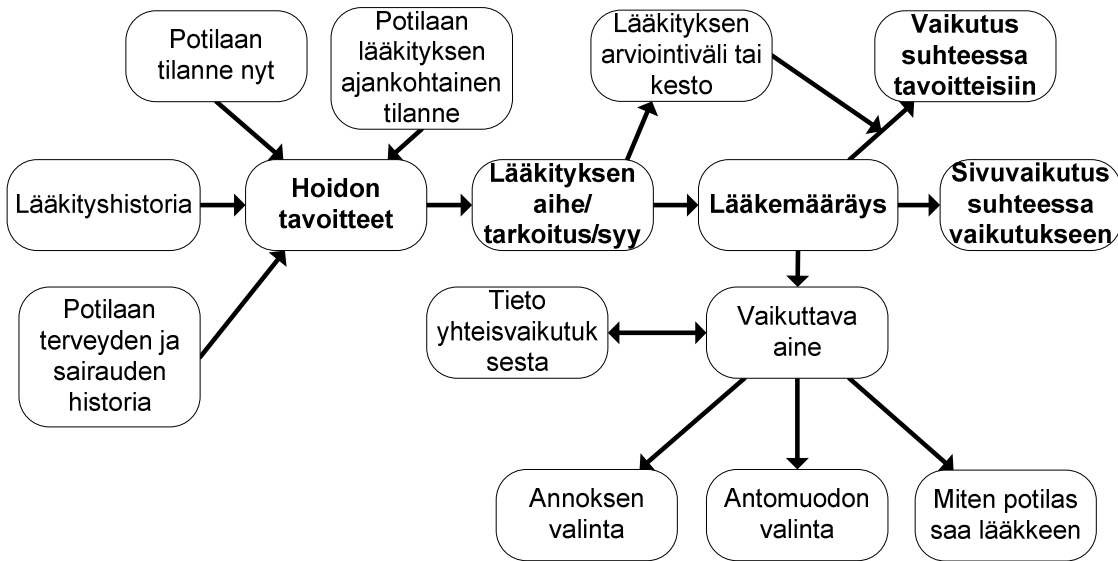
4) Lopetettu lääkitys (on ollut joko pitkäaikais- tai tilapäislääkitystä)

Lääkitys (jaottelu perustuen ydintietomäärittelyihin)			
1) Pitkäaikaislääkitys Geneerinen nimi Kauppanimi Indikaatio/Diagnoosi/Hoidon tavoite Koska aloitettu Kuka aloittanut Missä aloittanut Onko uusittu Kuka uusinu Missä uusittu Toteutunut annostelu Annostus eri aikaväleillä Hoitoprosessin vaihe Lääkityksen rooli (sairaudenhoito/työperäisen sairauden hoito/muu) Sairaala- vai kotilääke Lääkityksen arviointitarve Lääkityksen vaikutus Koetut sivuvaikutukset Sivuvaikutusten vakavuusaste Sivuvaikutusten varmuusaste Sivuvaikutuksen todennut lääkäri Sivuvaikutuksen todennut org	2) Pitkäaikaislääkitys (tarvittaessa käytettävä) Geneerinen nimi Kauppanimi Indikaatio/Diagnoosi/Hoidon tavoite Koska aloitettu Kuka aloittanut Missä aloittanut Onko uusittu Kuka uusinu Missä uusittu Toteutunut annostelu Annostus eri aikaväleillä Hoitoprosessin vaihe Lääkityksen rooli (sairaudenhoito/työperäisen sairauden hoito/muu) Sairaala- vai kotilääke Lääkityksen arviointitarve Lääkityksen vaikutus Koetut sivuvaikutukset Sivuvaikutusten vakavuusaste Sivuvaikutusten varmuusaste Sivuvaikutuksen todennut lääkäri Sivuvaikutuksen todennut org	3) Tilapäislääkitys (parhailaan käytössä) Geneerinen nimi Kauppanimi Indikaatio/Diagnoosi/Hoidon tavoite Koska aloitettu Kuka aloittanut Onko uusittu Kuka uusinu Missä uusittu Toteutunut annostelu Annostus eri aikaväleillä Kuri- vai kerta-annostelu Hoitoprosessin vaihe Lääkityksen rooli (sairaudenhoito/työperäisen sairauden hoito/muu) Sairaala- vai kotilääke Lääkityksen arviointitarve Lääkityksen vaikutus Koetut sivuvaikutukset Sivuvaikutusten vakavuusaste Sivuvaikutusten varmuusaste Sivuvaikutuksen todennut lääkäri Sivuvaikutuksen todennut org	4) Lopetettu lääkitys (pitkäaikais- tai tilapäislääkitystä) Geneerinen nimi Kauppanimi Koska aloitettu Indikaatio/Diagnoosi/Hoidon tavoite Kuka aloittanut Missä aloittanut Koska lopetettu Missä lopetettu Miksi lopetettu Kuka lopettanut Onko uusittu Kuka uusinu Missä uusittu Annostus eri aikaväleillä Toteutunut annostelu Hoitoprosessin vaihe Lääkityksen rooli (sairaudenhoito/työperäisen sairauden hoito/muu) Sairaala- vai kotilääke Lääkityksen arviointitarve Lääkityksen vaikutus Koetut sivuvaikutukset Sivuvaikutusten vakavuusaste Sivuvaikutusten varmuusaste Sivuvaikutuksen todentajan SV
Lääke: Geneerinen nimi Kauppanimi ATC-koodi Rinnakkaisvalmisteet Lääkemuoto Antomuoto Vahvuus Vaihdeavuus Vaikutus Koetut sivuvaikutukset Sivuvaikutusten vakavuusaste Sivuvaikutusten varmuusaste Sivuvaikutuksen todennut lääkäri Sivuvaikutuksen todennut org Lääkkeen riskitieto Annostus			

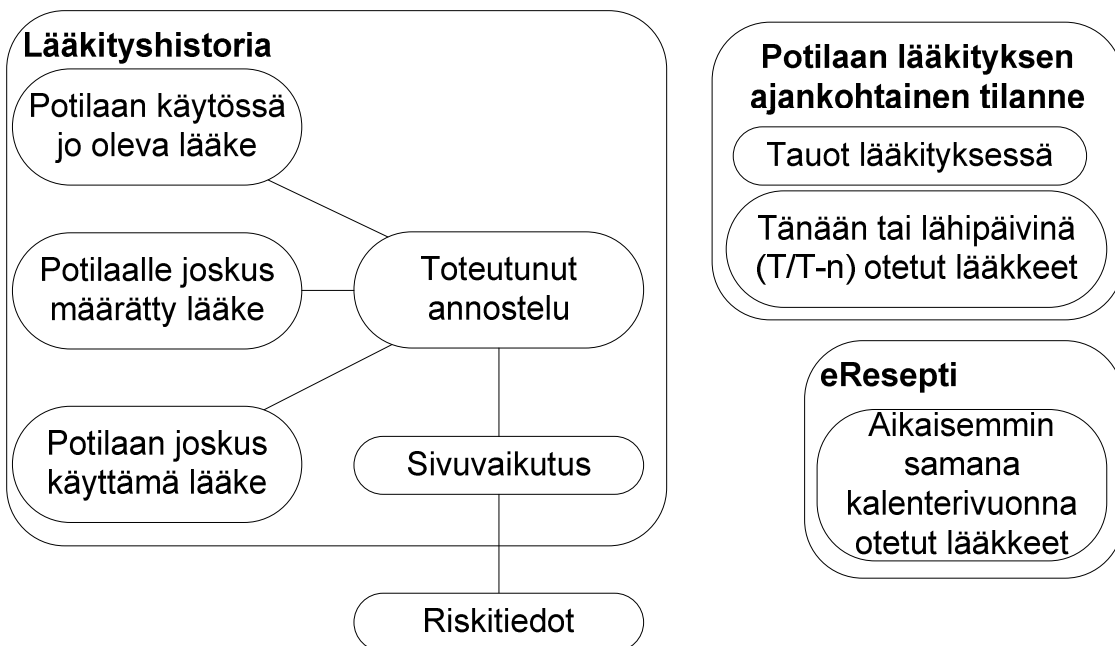
Kuva 1: Lääkitystiedon jaottelu ydintietomäärittelyjen mukaisesti [Nykänen ja muut, 2008]

2.2 Lääkitystieto terveydenhuollon ammattilaisen näkökulmasta

Kuvissa 2 ja 3 on kuvattu lääkitystiedon käyttöä ja tarvetta lääkärin näkökulmasta. Kuva 2 esittää lääkemääräyksen syntyprosessin, joka käynnistyy hoidon tavoitteen määrittämisestä – mihin tarkoitukseen lääkitystä tarvitaan. Kuvassa 3 on kolme eri teemaa miten lääkäri voisi tarvitsemansa tiedon jaotella: 1) Mitä lääkkeitä potilaalla on käytössä parhaillaan (ajankohtainen tilanne), 2) mitä hänelle on määrätty ja mitä lääkkeitä on noudeuttu (eResepti) ja 3) lääkityshistoria. Aiemmin käytetyt lääkitykset ovat voineet synnyttää uuden lääkkeen määräämisen kannalta tärkeitä tietoja, kuten sivuvaikutukset ja riskitiedot, joihin kuuluvat lääkeaineallergiat.



Kuva 2: Lääkitys ammattilaisen näkökulmasta [Nykänen ja muut, 2008, s. 36]



Kuva 3: Lääkärin tarvitsemat tiedot [Nykänen ja muut, 2008, s. 36]

Terveydenhuollon ammattilaisen lääkitystiedon tarpeiden kattamiseksi tulisi huomioida ainakin seuraavat asiat [Nykänen ja muut, 2008; Aarnio, 2008b]:

- Potilaan lääkeyshistoria mukaan lukien lääkeyksen vaikutukset ja sivuvaikutukset
- Lääkitystietojen kattavuus

- Lääkitystietojen kumuloituminen hoitoprosesseissa, hoitoon liittyvää lääkitystä voidaan tarkastella jatkumona
- Reseptin (sähköisen tai muun) tuottaminen kirjattujen tietojen perusteella
- Tiedot tarvitsee kirjata vain kerran, ei päällekkäistä kirjaamista
- Tietojen saatavuus tarpeen mukaan eri tilanteissa
- Lääkityksen hallinta eri ominaisuuksien perusteella: alkuperäisen määräyksen geneerinen nimi, lääkityksen indikaatio, lääkkeiden kauppanimet jne.
- Tietoja voidaan tarkastella voimassaolon, hoidon aiheen ja vaikuttavan aineen (geneerisen nimen) perusteella
- Yhdenmukaiset lääkitystiedot ja käsittelytavat eri järjestelmissä
- Ammatilaisen kirjaamat potilaan lääkitystiedot ovat saatavissa muuttumattomina toisen ammatilaisen käyttöön
- Lääkitystietojen harmonisointi ja lääkitysprosessien kokonaisvaltainen hallinta.

2.3 Lääkitystiedon hallintamallit

Lääkitystietoa voidaan esittää ja käyttää monin vaihtoehtoisin tavoin. Lääkitystiedon hallintamallien analysointi on yksi tapa arvioida, miten saavutettaisiin tilanne, jossa potilaan lääkitystiedot olisivat hyvin saatavilla ja hallittavissa. Lääkitystiedon hallintamalli on malli, joka nyt tai tulevaisuudessa soveltuisi terveydenhuollon eri organisaatioiden potilaita koskevien lääkitystietojen säilyttämiseen ja siirtämiseen [Aarnio, 2008b]. Mahdollisiin hallintamalleihin vaikuttaa ratkaisevasti toimintaympäristö, joka vaihtelee paitsi eri maiden välillä, mahdollisesti myös käytössä olevien potilastietojärjestelmien mukaan. Toimintaympäristöön vaikuttavat Aarnion [2008b] mukaan muun muassa:

- Sidosryhmät – Terveydenhuollon ammatilliset, julkiset ja yksityiset toimijat, kansalainen ja apteekki
- Teknologia – Nykyinen teknologia, erityisesti nykyisten potilastietojärjestelmien yhteensovittaminen, ja yleensä teknologian tuomat mahdollisuudet ja rajoitteet
- Lainsäädäntö – Lait mahdollistavat toiminnallisuutta, mutta voivat myös rajoittaa sitä
- Taloudelliset tekijät – Kaikista parhaaksi koettu vaihtoehto ei ole välttämättä taloudellisesti järkevä tai mahdollinen
- Päätöksentekoympäristö – Julkisten palvelujen tuottamisen malli [Denhardt and Denhardt, 2000], kansalaisen ja terveydenhuollon toimijoiden rooli lääkityksen ja hoidon hallitsijana.

Jotta lääkitystietoa voitaisiin käyttää tehokkaasti, lääkitystiedon hallintamallin tulisi mahdollistaa ainakin sellainen toiminnallisuus, että aiemmin esitellyt ammatilaisen

tarvitsemat lääkitystiedot ja toiminnot olisivat käytettävissä. Lisäksi hallintamallin tulisi tukea myös esimerkiksi apteekin, kansalaisen, vakuutusyhtiöiden ja Kelan tarpeita, jotta koko toimintaympäristö tulisi huomioiduksi. Aarnio [2008b] tarkastelee viittä vaihtoehtoista hallintamallia:

- Lääkitysriivi – Jokainen lääkkeen määräys on asiakirja, joka tallennetaan heti syntymisen jälkeen kansalliseen arkistoon. Lääkityksiä tarkastellaan yksittäisinä merkintöinä, joista voidaan muodostaa vaikuttavaan aineeseen perustuen esimerkiksi yksittäisiä lääkkeitä koskevia historiaketjuja tai listata voimassaolevat lääkkeet. Lääkityksistä muodostuu vain yksi lääkityslista
- Lääkityslista – Lääkitykset olisivat joko a) nykyiseen tapaan organisaatiokohtaisia listoja mahdollisesti rakenteettomassa muodossa tai siten, että b) organisaatiot käyttäisivät yhtä yhteistä listaa
- Lääkitys hoitokokonaisuutena
- Kansalainen oman lääkitystiedon korjaajana – kansalainen saisi jonkin laitteen tai palvelun avulla nähtäväkseen lääkitystietonsa, joista osaa hän voisi muuttaa
- eReseptiarkisto.

Malleista ei nostettu yhtä ylitse muiden, mutta selvää oli ainakin, että kauppanimi-perustaisesti hallintamallia ei voida rakentaa. Tärkeätä oli lisäksi, että lääkityslistoja olisi vain yksi – tällä hetkellä osa ongelman ydintä on epääjantasaiset ja puutteelliset dokumentit. Lääkityslistamalleissa on molemmissa pulmansa. Organisaatiokohtaisissa listoissa ongelmana on mahdolliset päällekkäisyydet ja epääjantasaisuus, yhteisen listan ongelma on se, että lääkityksiä ei voisi käsitellä yksitellen, jolloin niitä ei voi järjestellä ja tarkastella eri tavoin eikä kohdistaa niihin tiettyjä tapahtumia. [Aarnio, 2008b]

Parhaimmillaan hallintamalli olisi, jos kaikki viisi yllä olevaa mallia saataisiin yhdistettyä, koska niitä kaikkia tarvitaan, eikä mikään niistä ole yksin tarpeeksi joustava ja monipuolinen. Tällaisen hallintamallin voisi muodostaa kokonaisuus, jossa lääkitysriivi on yksittäinen lääkitykseen liittyvä merkintä, joista voidaan koostaa lääkityslista osastolle tai potilaalle. Samoja rivejä voitaisiin tarkastella hoitokokonaisuuksiin perustuen. Kansalaisen olisi hyvä voida tarkistaa lääkitystietonsa, mahdollisesti ammattilaisen avustuksella. eReseptiarkiston perusteella farmaseutti pystyy havaitsemaan monia potentiaalisia ongelmia lääkityksessä tai kehottaa uusimaan päättyvät reseptit. Lääkäri taas voisi tarkastella reseptien toimitustietoja.

KanTa-arkkitehtuurissa sähköiseen arkistoon tallennetaan potilasasiakirjoja. Laissa sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä [Laki 159/2007] määritellään, että palvelutapahtumalla tarkoitetaan ”terveydenhuollon palvelujen antajan ja potilaan välistä yksittäisen palvelun järjestämistä tai toteuttamista” ja palvelukokonaisuudella ”yhden tai useamman terveydenhuollon palvelujen antajan tuottamien palvelutapahtumien yksilöityä kokonaisuutta”. Palvelutapahtuma liittyy aina johonkin palvelukokonaisuuteen, joka voi olla lyhyt tai pitkä ja voi koskea yhtä hyvin pitkäai-

kaista hoitoa tai kertaluonteista antibioottikuuria. Lääkitystietojen tarkastelun ja ajantasaisuuden ylläpitämisen mahdollisuuksiin tuo haasteen se seikka, että arkistosta tietoa saa vain tarkasteltavaksi, sitä ei voi yhdistellä oman organisaation tietoihin ja tallentaa. Nykyisten määritysten ja lakien rajoissa KanTa-arkistosta ei ole saatavissa eikä sen avulla voi tuottaa lääkityslistaa, johon oman organisaation järjestelmä automaattisesti yhdistelisi oman ja muiden organisaatioiden potilaalle määräämät lääkitykset ja lääkityksen muutokset. On tietysti huomioitava, että joka tapauksessa tällaisen mekanismin toteuttaminen olisi varsin suuri urakka ja ensimmäisten kansallisten palveluiden käyttöönotto on vasta alkamassa.

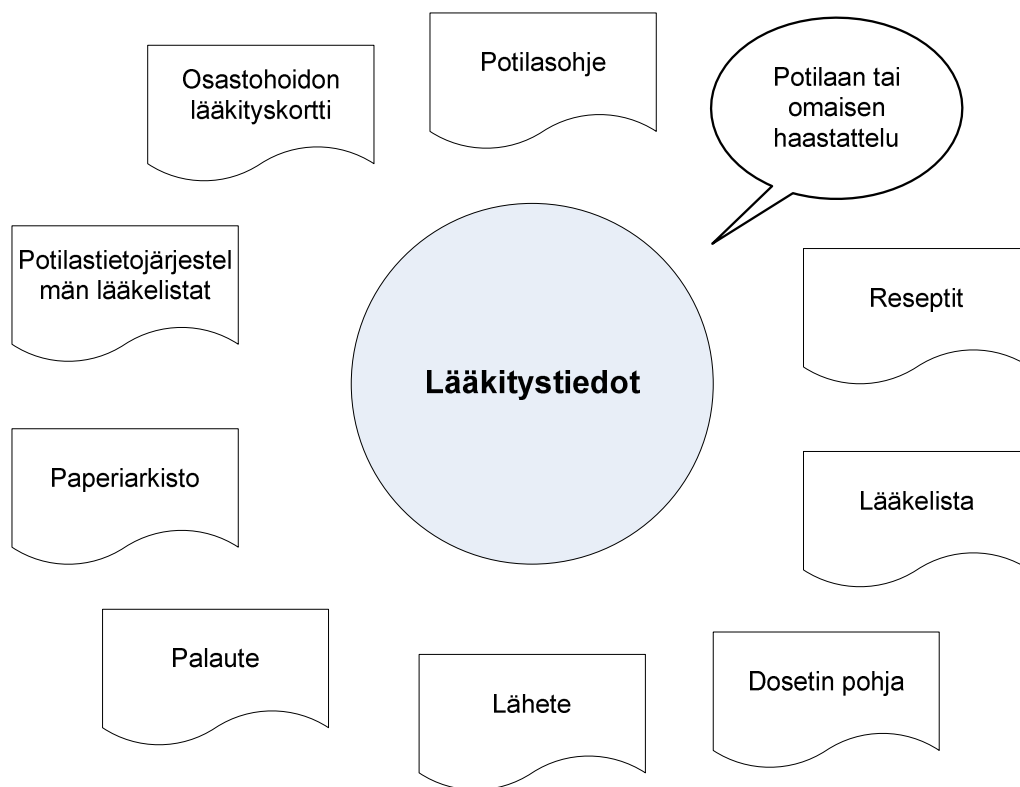
Potilaan lääkitystä olisi voitava tarkastella jatkumona, joka kulkee kitkattomasti saumakohtien – palvelutapahtumien ja palvelukokonaisuuksien – yli. Lääkitystä pitäisi voida tarkastella useasta näkökulmasta, ei pelkästään kaupanimen tai vaikuttavan aineen perusteella. Lääkityksen voisi ryhmitellä esimerkiksi hoidon aiheen perusteella. Potilas voi syödä vuosikymmeniä lääkkeitä samaan vaivaan, ja lääkityshistoriaa olisi luonnollista tarkastella hoidon aiheen näkökulmasta. Esimerkiksi vaikuttava aine tai kauppanimi on hyvin voinut vaihtua useasti hoitohistorian aikana, jolloin vain näiden ominaisuuksien perusteella lääkityshistoriaa tarkasteltaessa nähdään päättyneitä ja alkaneita lääkityksiä, jotka eivät ole yhteydessä toisiinsa. Samoin jos lääke jää hetkeksi tauolle ja kokeillaan toista lääkettä, puuttuu konteksti, joka kertoisi, että tauolla ja uudella lääkkeellä on yhteys. Asia selviää yleensä potilaskertomusta lukemalla, mutta huomattavasti luotettavampaa ja käytännöllisempää olisi, jos lääkityksiä voisi suoraan tarkastella hoidon aiheen perusteella.

2.4 Lääkitystiedon lähteet

Lääkitystietoa syntyy useassa eri tilanteessa ja sitä myös käytetään monin eri tavoin. Sen pohjalta tuotetaan monenlaisia dokumentteja ja niillä on erilaisia käyttötarkoituksia. [Nykänen ja muut, 2008; Aarnio, 2007b] Kuva 4 kokoo keskeisiä lääkitystiedon lähteitä ja käyttötapoja, joita lääkäri voi tarvita ja joiden pohjalta muodostuu käsitys kokonaislääkityksestä. Eri tilanteissa tarvitaan erilaisia asiakirjoja, eikä niiden määrä itsessään ole ongelma. Sen sijaan ongelmia syntyy silloin, jos asiakirjoissa on eri tiedot, eikä ole varmuutta tiedon paikkansapitävyydestä, onko kädessä varmasti uusin lääkekortti vai onko uudempikin tuotettu. Se, mitä tietoa lääkäri tarvitsee, riippuu muun muassa hoidon aiheesta, potilaan aiemmasta hoidosta, onko hän ollut osastohoidossa lähiaikoina tai onko hän käynyt yksityisellä vastaanotolla. Lääkäri saa oman organisaationsa sähköisistä järjestelmistä kyseisessä organisaatiossa kirjatut lääkitystä koskevat tiedot, mutta potilaan muissa organisaatioissa saamasta hoidosta ja lääkityksestä tietoa on saatavilla sähköisenä vaihtelevasti riippuen muun muassa siitä, millainen aluetietojärjestelmä on käytössä ja onko hoidosta saatu sähköinen palaute. Usein tietoa kulkee potilaan mukana paperilla esimerkiksi jos potilas siirtyy erikoissairaanhoidosta terveyskeskuksen

osastolle. Ei ole tavatonta, että ajankohtainen lääkitystieto kerätään dosetin pohjassa olevasta listasta [Aarnio, 2008a].

Potilas on ammattilaiselle keskeinen tiedonlähde, koska omaisen ohella hän on usein ainoa, joka voi vastata kysymyksiin lääkkeiden syömisestä, vaikutuksista ja sivuvaikutuksista. Sähköisen reseptin myötä lääkäri voi tulevaisuudessa saada tiedon siitä, mitä lääkkeitä potilas on hakenut apteekista [STM, 2008; HE, 2006], mutta vain potilas itse tietää, onko lääkkeet syöty tai onko ohella käytetty myös esimerkiksi käsikauppalääkkeitä, joista reseptiä ei ole tai onko potilas saanut paperireseptejä muualta. Lääkärin ja potilaan välinen vuorovaikutus on jatkossakin keskeinen väline erityisesti ajankohtaisen tilanteen osalta lääkitystiedon saamisessa ja antamisessa.



Kuva 4: Lääkitystiedon lähteet [Nykänen ja muut, 2008; Aarnio, 2007b; Aarnio, 2008a]

2.5 EU:n suositukset potilaan lääkitystiedoiksi

EuroRec (European Institute for Health Records) on tuottanut määräykset toiminnoista ja ominaisuuksista, joita tarvitaan lääkitystiedon hallitsemiseksi [EuroRec, 2006]. Lista (Taulukko 1) voi toimia hyvänä lähtökohtana potilastietojärjestelmää kehitettäessä tai arvioitaessa. Lääkitystiedon tietoaarkkitehtuurin kontekstissa se toimii kriteeristönä tie-

tomallin ja tietoarkkitehtuurin kattavuuden varmistamisessa. Suomessa sekä EU:n alueella käytössä olevat potilastietojärjestelmät täyttävät kriteereistä vaihtelevan määrän [Mäkinen ja Nykänen, 2008].

Taulukko 1: EuroRec-määrittelyt lääkitystiedoille [Eurorec, 2006].

Keyword	Criteria
Medication list	The system shall provide the ability to create and maintain medication lists.
Show medication	The system shall display the current active medication for each patient.
Prescribing – information	The system shall record the prescribing of medication, including the indication/diagnosis.
History	The system shall provide full history for change in the medication (eq. Dates for start, ordering, modifications, renewal and end).
Update of medication table	The system shall allow for amendments to the drug/medication database, defined by the user.
Print and reports	The system shall provide the ability to print all the medication information, which is stored in the system (reports and prescription using actual legal form).
Guidelines and procedure instructions	The system shall provide access to guidelines and procedure instructions that can be customised by the health organisation.
Prescribing – valid information	The system shall check that the prescribed data are valid and include sufficient information for administration and the delivery at the pharmacy.
Generic medication	The system shall maintain a link between specific medication products and the equivalent generic products.
Search	The system shall provide good capabilities for searching medication, including search on specific and generic products.
Coded information – list of medication	The system shall maintain a coded list of medication
Prescribing – print	The system shall provide the ability to print

	and re-print prescriptions.
Prescriptions – electronic communication	The system shall provide the ability to send prescription electronically.
Prescription – decision support	The system shall display a dose calculator for patient-specific dosing, based on weight and age.
Coded information – update	The system shall provide the ability for regular medication update including drug interaction databases.
Medication list	The system shall provide the ability to create provider specific medication list of the most commonly prescribed drugs with specific attributes eq. A default dose, frequency and quality.
Prescribing – price	The system should list of generic medication sorted by the cheapest product first
Administration	The system shall provide the ability to document medication administration.
Prescribing – predefined	The system shall allow defining preferred prescriptions for use in certain conditions.
Medication – magisterial drugs	The system shall provide the ability for magisterial preparation and recording.

Eurorec-suositus on melko hyvä referenssi kattaen muun muassa lääkitystiedon koodituksen, potilaan ajankohtaisen lääkityksen, lääkityshistorian, muutokset lääkityksessä sekä lääkkeiden annostelun. Siinä on kuitenkin joitakin seikkoja, jotka tarvitsisivat täsmennystä ainakin puhuttaessa Suomea koskevista toteutuksista. Ainakaan taulukossa esitetyllä abstraktiotasolla esitettynä määrittely ei sisällä ydintietomäärittelyiden mukaista luokittelua pitkäaikaisiin, määräaikaisiin, tarvittaessa otettaviin ja lopetettuihin lääkityksiin.

Toinen merkittävä rajoite on lääkitystiedon ja reseptitiedon sekoittuminen. Aiemmin toin esiin, että lääkitystiedon ja reseptin välille ei tulisi vetää yhtäläisyysmerkkejä, koska resepti eli lääkemääräys on ensisijaisesti dokumentti lääkkeen toimittamista varten, vaikka sen sisältämiä tietoja voidaan käyttää muuhunkin. Mielestäni lääkitystietoa ei tulisi ensisijaisesti tarkastella potilaan saamiin resepteihin perustuen. Potilastietojärjestelmään kirjataan toki monenlaista resepteihin liittyvää lääkitystietoa, mutta ne ovat vain osa lääkitystietoa. Suomessa sähköisen reseptin tietosisältö on määritelty omana

kokonaisuutenaan [STM, 2008]. Reseptikeskus on tarkoitettu pääasiassa reseptien tallentamiseen ja käsittelyyn, ei lääkitystiedon hallinnan työkaluksi.

Lääkitystietoon tulisi yhdistyä myös tieto lääkeaineallergioista sekä lääkkeiden mahdollisesti aiheuttamista sivuvaikutuksista, jotka voivat olla esimerkiksi lääkkeen annostuksen muutoksen tai lopettamisen syynä. Taulukosta 1 ei selviä, tulevatko nämä huomioituiksi, mutta ne voivat hyvin olla määritettynä johonkin toiseen osioon samoin kuin ydintietomäärityksissä [STM, 2007b] on riskitiedot. Raja lääkitystiedon ja potilasta koskevan muun tiedon välillä on usein häilyvä, mutta allergioilla ja sivuvaikutuksilla tulisi olla selvä yhteys tiettyyn lääkemääräykseen.

3 Tutkimusongelma, rajaukset ja tutkimusmenetelmät

3.1 Tutkimusongelma ja sen rajaus

Lääkitystiedon hallinta -hankkeessa on tuotettu lääkitystiedon prosessi- ja tietomallit. Prosessimallien ja muun materiaalin pohjalta tuotetussa tietomallissa kuvataan lääkärin näkökulmasta eri hoitoprosessin vaiheissa käytettävät tiedot. Kuvaus on kattava ja johdonmukainen, eikä takerru tarpeettomasti teknisiin yksityiskohtiin ja se on sopiva tutkimukseni perustaksi.

Tässä tutkimuksessa johdan lääkitystiedon tietomallin sekä muun lähdemateriaalin pohjalta lääkitystiedon tietoarkkitehtuurin. Tietomallissa on tunnistettu tietotarpeet, määritelty yhteinen tietosisältö ja tiedon rakenne.

Tutkimuksen fokus on lääkitystiedon tietoarkkitehtuurin kuvaamisessa terveydenhuollon ammattilaisen, erityisesti lääkärin näkökulmasta. Apteekin tai potilaan näkökulma lääkitystietoon ja sähköinen resepti rajautuvat tämän tutkimuksen ulkopuolelle. Sähköiselle reseptille on jo toteutettu omat tietosisällön määrytykset ja toiminnalliset kuvaukset [STM, 2008]. Edellä mainittujen näkökulmien yhteyksiä tietoarkkitehtuuriin voidaan kuitenkin tarvittaessa sivuta. On myös paljon tietoa, jonka ainoastaan potilas tai omainen voi kertoa lääkityksestä, kuten sen haittavaikutukset tai onko lääkkeitä syöty. Potilaan ja kansalaisen näkökulma jää kuitenkin tämän tutkimuksen ulkopuolelle, koska ne eivät ole olleet osa prosessi- ja tietomallinnusvaihetta.

Tietoarkkitehtuuri kuvataan yleisellä tasolla, eikä siinä analysoida organisaatioiden sisäistä toimintaa, kuten sairaalan, terveydenhuollon organisaation tai osastohoidon sisäistä tiedonvälitystä, tietojärjestelmiä ja niiden sisäistä tietoarkkitehtuuria. Se on geneerinen tekemisen tavasta ja teknologiasta riippumaton kuvaus, johon eivät kuulu ne työprosessit, tietojärjestelmä- tai teknologiaratkaisut, joilla sen kuvaama tieto tuotetaan tai tallennetaan. Tietoarkkitehtuurin antaa tietomallille harmonisoidun rakenteen, joka on yhtenevä valittuihin sovellusalueen viitemalleihin.

3.2 Tutkimusmenetelmät

Tutkimuksessa perehdytään erityisesti terveydenhuollon tietoarkkitehtuureihin liittyviin malleihin ja sovelletaan valittuja malleja lääkitystiedon tietoarkkitehtuurin suunnitteluun. Kyseessä on suunnittelutieteellinen tutkimus.

Marchin ja Smithin [1995] sekä Hevnerin ja muiden [2004] mukaan suunnittelutieteellisessä tutkimuksessa saadaan neljänlaisia tuloksia: käsitteistöjä (constructs), malleja (models), metodeja (methods) ja toteutuksia (instantiations) eli artefakteja. March ja Smith [1995] määrittelevät luonnontieteellisen ja suunnittelutieteellisen tutkimuksen

eroavan toisistaan siten, että ensimmäinen pyrkii tulkitsemaan todellisuutta, asioita ja ilmiöitä, kun taas jälkimmäinen suunnittelee ihmisten tarpeista kumpuavia toteutuksia. Suunnittelutieteellisen tutkimuksen lopputulosta voidaan arvioida kysymyksen: Toimii-ko se? Onko se parempi kuin aiemmat toteutukset? Suunnittelutiede on teknologia-orientoitunutta ja siksi se on helppo mieltää esimerkiksi jonkin konkreettisen tuotteen suunnitteluksi ja toteuttamiseksi, mutta Järvinen ja Järvinen [2004] laajentavat toteutuksen tarkoittamaan myös kaikenlaisia tavoitteisia innovaatioita.

Järvinen ja Järvinen [2004] puhuvat suunnittelutieteiden yhteydessä innovaatioiden toteuttamisesta. Näin usein onkin, sillä tavoitteena olevan artefaktin motivaationa on joko se, että sellaista ei vielä ole tai sen toteutus on nykyisin puutteellinen. Tämän tutkimuksen lähtötilana on tietotarpeita vastaavan lääkitystiedon tietoarkkitehtuurin puute ja tavoitteena lääkitystiedon käsitteiden ja mallien määrittely sekä tietoarkkitehtuurin toteutusmetodin suunnittelu. Tutkimuksessani ei edetä varsinaisen toteutuksen asteelle, vaan kyseessä on suunnitelma siitä, miten lopullinen tietoarkkitehtuuri käytännössä voidaan toteuttaa.

Tutkimuksen teoriapohja perustuu ennen kaikkea kirjalliseen aineistoon. Lähteinä on kansallisia raportteja ja määrittämiä, terveydenhuollon viitemalleja, viitekehyksiä ja standardeja, tieteellisiä artikkeleita sekä Lääkitystiedon hallinta -hankkeessa tuotettu dokumentaatio, jonka taustalla on runsaasti tiedonkeruuta. Tietoarkkitehtuurin syntyprosessiin kuuluu tyypillisesti haastatteluita tai muita vastaavia tiedonkeruumenetelmiä. Taustatyö on kuitenkin tehty lääkitystiedon tietomalliin asti, joten näille tiedonkeruumenetelmille ei ole enää tässä tutkimuksessa tarvetta.

3.3 Mallinnusmenetelmät

Prosessi on sarja toimintoja, jotka muuttavat panokset tuotoksiksi [Karimaa, 2001]. Sillä on alku- ja lopputila, joiden välillä tapahtuu sarja muutoksia – prosessin vaiheita – jotka muuttavat panokset tuotoksiksi. Prosessin resursseja ovat systeemin muodostavat ihmiset, koneet, laitteet, toimitilat ja tiedot. Prosessin panokset saadaan näiltä resursseilta, ja se tuottaa lisäarvoa, jos tuotos on suurempi kuin käytetyt panokset [Karimaa, 2001].

Prosessimallia käytetään kuvaamaan prosessin kulku. Prosessia voidaan kuvata eri näkökulmista, kuten asiakkaan, työntekijän, johdon tai tietojärjestelmän näkökulmasta. Prosessin voi kuvata myös eri abstraktiotasoilla ja siinä voi olla aliprosesseja. [Mäkinen and Nykänen, 2009]

Tietomallinnus on tärkeimpiä tietojärjestelmän suunnittelun vaiheita. Prosessi- tai tietovirtamalleilla voidaan kuvata miten, missä ja milloin tietoa käytetään tai tallennetaan, mutta tietomalli tarvitaan määrittelemään tieto, sen rakenne ja suhteet.

Käsitteellinen tietomalli on mahdollisimman tarkka kuvaus kohdealueen tiedosta. Se kuvaa toimintaan liittyvän tiedon ja selittää entiteettien ja niiden attribuuttien välisiä

suhteita, merkityksiä ja sääntöjä. Kuvaus on riippumaton käytetyistä tietokanta- ym. ratkaisuksista. [Hoffner et al., 1999]

Hyvin yleinen tietomallinnusmenetelmä on ER-mallinnus (Entity-Relationship) [Chen, 1976]. Se rakentuu kohteista eli entiteeteistä, niiden välisistä suhteista ja näihin molempiin liittyvistä ominaisuuksista eli attribuuteista. Se on yksityiskohtainen, looginen kuvaus kohdealueen tiedosta.

Kohde on esimerkiksi henkilö, paikka, tapahtuma, käsite tai objekti. Kohteen instanssi on kohteen ilmentymä. Kohteen ilmentymiä voi olla useita, esimerkiksi työntekijöitä yrityksessä. Saman kohteen ilmentymien ominaisuudet ovat samat, mutta ominaisuuksien arvot voivat olla erilaiset, esimerkiksi työntekijän nimi.

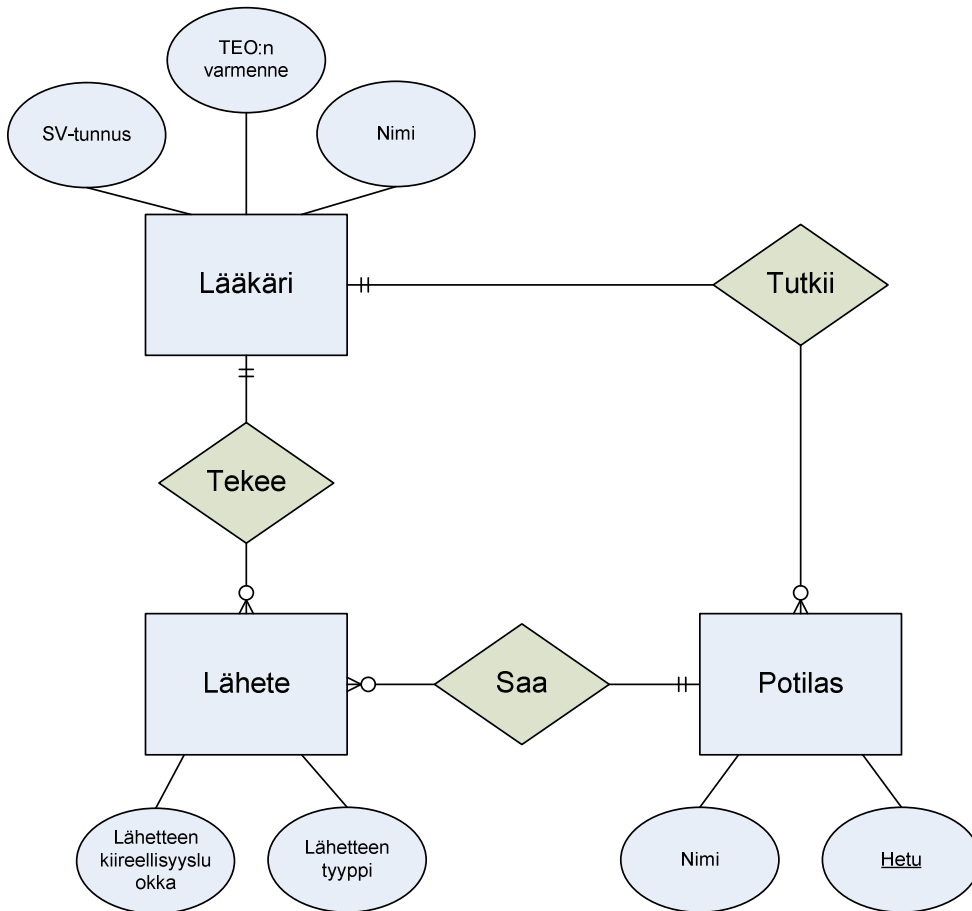
Kohteiden välinen suhde eli relaatio tarkoittaa, että kohteiden välillä on jokin luonnollinen suhde. Sitä kuvataan tavallisesti verbillä, esimerkiksi potilas *saa* lähetteen (Kuva 5). Verbin valinnassa on hyvä miettiä sopivan laaja kuvaus, muutoin voidaan tarpeettomasti rajata keinovalikoimaa. On parempi sanoa lääkäri *tutkii* potilaan, kuin lääkäri *tapaa* potilaan, jotta ei rajata pois sitä vaihtoehtoa, että lääkäri tekee lähetteen vain potilaasta saamiensa tietojen perusteella. Aina suhteen esittämiseen ei löydy täysin luontevaa verbiä, jolloin voidaan tyytyä sanomaan esimerkiksi *liittyy*.

Suhteella on myös kardinaliteetti, joka kertoo montako kohteen ilmentymää voi tai täytyy liittyä toiseen kohteeseen. Kuvan 5 esimerkissä yksi lääkäri tutkii 0–n potilasta. Samoin lääkäri voi tehdä 0–n lähetettä ja potilas voi saada 0–n lähetettä. Kukin lähete koskee tasan yhtä potilasta. Kardinaliteetin voi määrittää valitun näkökulman perusteella. Mikäli potilas olisi otettu keskiöön, kaavio voitaisiin kääntää siten, että yhdellä potilaalla on 0–n lääkäriä, joista kukin määrää 0–n lähetettä. Tällöin voisi myös harkita verbin *tutkii* vaihtamista.

ER-malli ei pysty kuvaamaan instanssien ajallista järjestystä eikä asettamaan loogisia ehtolauseita. ER-malli sallii, että lähete voisi yhtä hyvin syntyä ennen kuin lääkäri on tutkinut potilaan tai lääkäri voisi tutkia useita potilaita samanaikaisesti.

Kohteeseen liittyy ominaisuuksia, joista kuvassa 5 on annettu joitakin esimerkkejä, kuten potilaan nimi ja henkilöturvautunnus (Hetu). Kohteen instanssit voidaan erotella avainominaisuuden perusteella toisistaan, ja jokaisella kohteella tulee olla niitä ainakin yksi, potilaalla se voisi olla Hetu ja se on kuvassa alleviivattu. Avainominaisuuden on oltava sellainen, että kahdella ilmentymällä ei voi olla samaa arvoa tai jos kohteella on useita avainominaisuuksia, niistä ei voi syntyä samanlaisia yhdistelmiä.

Kuvassa 5 ER-mallin kuvaamiseen on käytetty yleistä notaatiotapaa, jossa kohteet kuvataan suorakulmioina ja kohteen yhdistävän suhteen verbi salmiakkina. Ominaisuudet kuvataan ovaaleina, jotka liittyvät johonkin kohteeseen. Entiteettien välisen suhteen kardinaliteetin kuvaukseen on käytetty tässä esimerkissä ns. crow's foot notaatiota.



Kuva 5: ER-mallin notaatio

UML (Unified Modeling Language) on paljon käytetty monipuolinen ja standardoitu mallinnuskieli, joka on syntynyt tietojärjestelmä- ja ohjelmistokehityksen tarpeisiin. Monikäyttöisyyttä tukee se, että UML on analyysi- ja mallinnusmetodologioista riippumaton. Sitä käytetään tyypillisesti liiketoiminnan mallintamiseen ja ohjelmistosuunnitteluun, mutta potentiaalisia käyttökohteita on monia muitakin. UML:ssä on kolme erilaista kaaviota, jotka jaetaan kolmeen kategoriaan [OMG, 2009]:

- Rakennekaavio (Structure diagram): komponenttikaavio, koostekaavio, luokkakaavio, oliokaavio, pakkauskaavio ja sijoittelukaavio
- Vuorovaikutuskaavio (Interaction diagram): ajoituskaavio, kokoava vuorovaikutuskaavio, kommunikaatiokaavio ja sekvenssikaavio
- Käyttäytymiskaavio (Behavior diagram): käyttötapauskaavio, tilakaavio ja aktiviteettikaavio.

4 Tutkimuksen taustateoriat, mallit ja viitekehukset

4.1 Kokonaisarkkitehtuuri

Schekkerman [2004] määrittelee järjestelmän kokonaisarkkitehtuurin (Enterprise Architecture, EA) olevan kattava, korkean tason kuvaus kohdealueesta. Se on kohdealueen yleiskaava, yhteistyön väline eri näkökulmien välillä. Näkökulmia voivat olla esimerkiksi liiketoiminnan suunnittelu (päämäärät, visiot, strategiat), liiketoiminta (termit, organisaatorakenne, prosessit ja data), automaatio (tietojärjestelmät ja tietokannat) ja teknologia-infrastruktuuri (tietokoneet, käyttöjärjestelmät ja tietoverkot).

Kokonaisarkkitehtuuria voi hyvin verrata kaupunkisuunnittelun yleiskaavaan, jossa kuvataan esimerkiksi teollisuuden, liikenteen ja asuinalueiden sijoittumista, mutta niiden tarkkaa toteutusta ei vielä määritetä. Kun tällainen kaava on laadittu, voidaan kaupunkia kehittää johdonmukaisesti. Yleiskaava tuo ennustettavuutta ja sen avulla voidaan koordinoita eri osapuolien (vrt. näkökulmien) tekemisiä. Esimerkiksi yhdyskuntatekniikan suunnittelussa tieverkostoa on tarpeen tietää, mihin on kaavoitettu asuin- tai teollisuusalueita sekä minkä luonteisia ne ovat. Teollisuusalueelle voidaan tarvita leveämmät tiet kuin asuinalueelle ja tilaa täytyy ehkä jättää kiskoillekin. Jos kaupunkia kehitettäisiin vain pienissä palasissa ilman yleissuunnitelmaa, infrastruktuurin rakentaminen perustuisi liiaksi arvailuun, jonka seuraukset voivat näkyä vasta paljon myöhemmin. Schekkerman [2004] tiivistää kokonaisarkkitehtuurin tehtävän olevan tulevaisuuden mahdollistaminen, ei sen ennustaminen. Yleiskaava voi kantaa kymmeniä vuosia eteenpäin – jos kaavaan ei ole jätetty tilaa uudelle radalle tai ohitustielle, tilanteen korjaaminen on hyvin hankalaa siinä vaiheessa, kun metsän tai pellon tilalla onkin asuinalue.

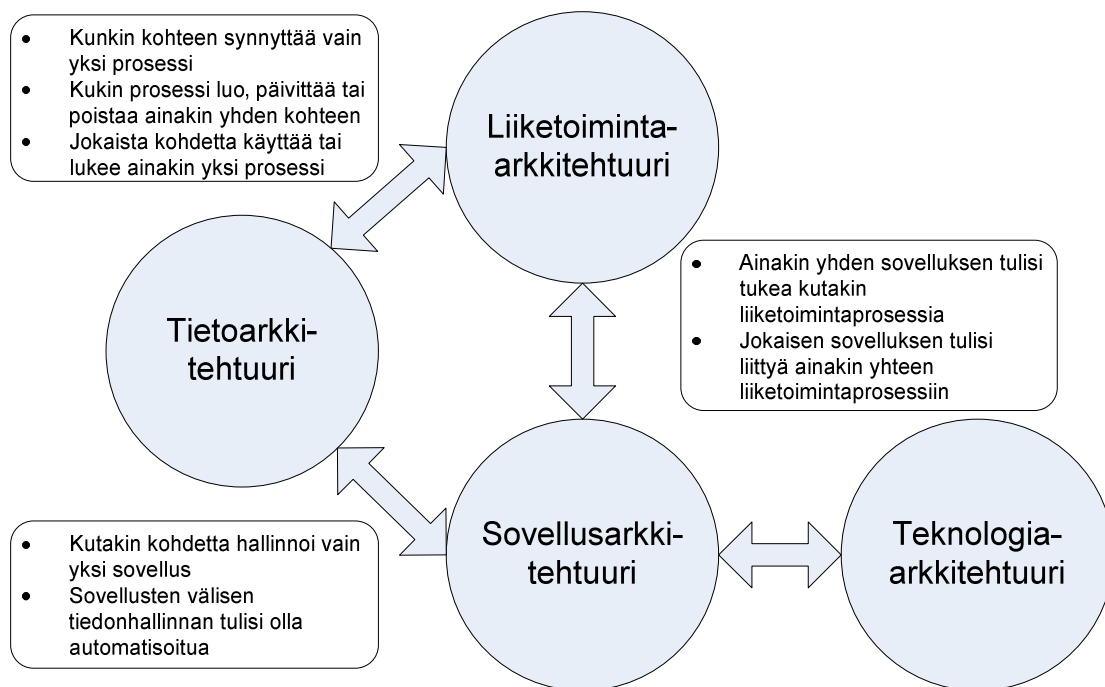
Schekkermanin [2004] kokonaisarkkitehtuurin tasojaottelussa informaatiotaso muodostaa selkeästi oman kokonaisuutensa suhteessa muihin tasoihin:

- **Liiketoimintataso**
Liiketoiminnan rakenteet, niiden väliset suhteet, tehtävät ja aktiviteetit.
- **Informaatiotaso**
Määrittää keskeiset tietovirrat ja tiedon ominaispiirteet.
- **Tietojärjestelmätaso**
Määrittelee tarvittavat ratkaisut, funktiot, ominaisuudet ja standardit.
- **Teknologiataso**
Määrittelee tarvittavat teknologiset palvelut, mutta ei toteutustapaa.

Kuvan 6 malli kuvaa kokonaisarkkitehtuurin tasojen välisiä suhteita sekä tasojen välisiä sääntöjä [Pereira and Sousa, 2003; Vasconcelos, 2004]. Siltanen [2004] on pro gradu -tutkielmassaan täydentänyt mallia lisäämällä teknologia-arkkitehtuurin malliin mu-

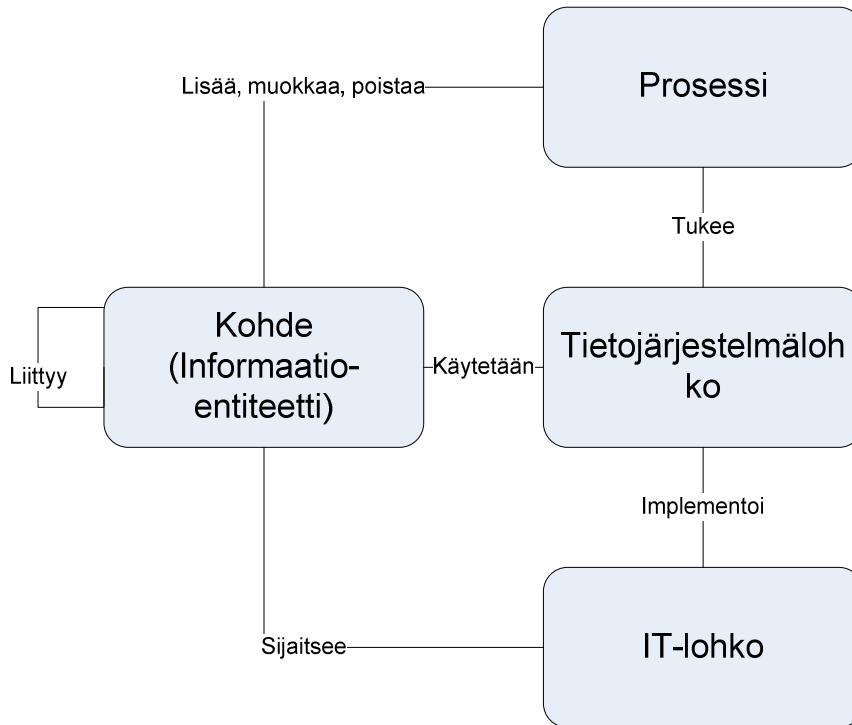
kaan, minkä jälkeen se vastaa Schekkermanin [2004] kokonaisarkkitehtuuria lukuun ottamatta sitä, että tietojärjestelmäarkkitehtuurin sijaan mallissa puhutaan sovellusarkkitehtuurista.

Mallin (Kuva 6) säännöt kuvaavat ideaalitulannetta, jossa arkkitehtuurin osat ovat tasapainossa. Tasapaino arkkitehtuurin tasojen välillä on Pereira ja Sousan [2003] mukaan toiseksi tärkein tietojärjestelmän onnistumistekijä. Hyvin tavallinen tilanne kuitenkin on, että yrityksen sovellus- ja liiketoiminta-arkkitehtuuri ovat epätasapainossa; että liiketoiminnan prosessien ja käyttäjien tarpeet ja tietojärjestelmän toteutus eivät kohta.



Kuva 6: Kokonaisarkkitehtuurin tasojen väliset suhteet mukaillen [Vasconcelos et al., 2004; Siltanen, 2004; Pereira and Sousa, 2003]

Kokonaisarkkitehtuurin tasojen välistä vahvaa riippuvuutta voi kuvata Vasconcelosin ja muiden [2003] CEO tietojärjestelmäarkkitehtuurin metamallin pohjalta, jota olen mukaillut tutkimukseni tarpeisiin kuvassa 7. Mallissa on nähtävissä tiedon suhde muihin arkkitehtuurin tasoihin. Prosessissa käsitellään tietoa tietojärjestelmän tukemana, jota varten puolestaan tarvitaan teknologinen implementaatio. Tieto sijaitsee jossakin teknologisessa tietojärjestelmän osassa, vaikkapa tietokannassa tai XML-tiedostossa.



Kuva 7: Arkkitehtuuritasojen väliset suhteet mukailten Vasconcelosin ja muiden [2003] CEO-tietojärjestelmäarkkitehtuurien metamallia.

4.2 Arkkitehtuuriviitekehys

Arkkitehtuuriviitekehys ohjaa ja tukee kokonaisarkkitehtuurin tuottamista ja hallitsemisesta. Arkkitehtuuriviitekehys on kommunikaation malli tietojärjestelmän kokonaisarkkitehtuuria kehitettäessä [Schekkerman, 2004]. Se voi sisältää erilaisia työkaluja, menetelmiä, periaatteita, lähestymistapoja, visualisointeja, standardeja ja muita asioita mitä tietojärjestelmäarkkitehdin tarvitsevia asioita.










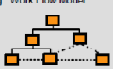


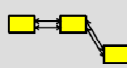

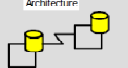
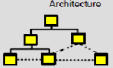





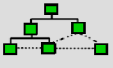








4.2.1 Zachmanin viitekehys

Zachmanin [1987; Schekkerman, 2004] arkkitehtuuriviitekehys on laajalti tunnettu kokonaisvaltainen malli, jolla kuvataan kohdealueen informaatioinfrastruktuuri suunnittelijan (planner), omistajan (owner), järjestelmäsuunnittelijan (designer), toteuttajan (builder), alihankkijan (subcontractor) ja toimivan järjestelmän (functioning enterprise) näkökulmista. Useita näkökulmia käyttämällä varmistetaan siitä, että kohdealue ja asioiden väliset suhteet on kuvattu kattavasti ja selkeästi. Malli on luonteeltaan proseduraalinen, mutta varsinaista estettä ei ole sen käytölle muulla tavoin, kuten olioperustaiseen suunnitteluun. Moni arkkitehtuuriviitekehys haarautuu Zachmanin viitekehystä erilaisin painotuksin. Esimerkiksi IAF (Integrated Architecture Framework) perustuu Zachmanin viitekehukseen, mutta jakautuu vertikaalisesti neljään tasoon: liiketoiminta-,

informaatio-, tietojärjestelmä- ja teknologiainfrastruktuuritaso, kuten myös E2A (Extended Enterprise Architecture).

Zachmanin viitekehysten kantavia ajatuksia on, että valmis ja todenmukainen kohdealueen kuvaus voidaan tuottaa kuvaamalla kohdealue kullakin arkkitehtuuritasolla kuuden kysymyksen näkökulmasta: mitä, miten, missä, kuka, milloin ja miksi. Näkökulmat ovat abstraktioltaan erilaisia ja siten vähentävät yksittäisten mallien kompleksisuutta. Arkkitehtuurin suunnittelun tasoja ovat laajuus, liiketoimintamalli, järjestelmämalli, teknologiamalli ja lopulta yksityiskohtaiset kuvaukset kohdealueesta. Prosessin tuloksena saadaan kattava kuvaus yrityksen tiedoista, toiminnasta, verkostosta, organisaatiosta, aikataulusta ja strategiasta. (Kuva 8)

Tietoarkkitehtuuri kohdistuu tietoon ja tiedon rakenteeseen, ja sen suunnittelu sijoittuu viitekehyksessä liiketoimintamallin tasolle – tai ehkä vielä tarkemmin liiketoimintamallin ja järjestelmämallin välimaastoon. Mitä kuvaa tietoarkkitehtuurin tiedon ja missä, miten, kuka, milloin ja miksi ohjaavat arkkitehtuurin rakenteen muodostumista. Tietoarkkitehtuuri siis kuvaa tietoa (mitä), muttei sen käyttöä (miten).

ENTERPRISE ARCHITECTURE - A FRAMEWORK TM							
	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>	
SCOPE (CONTEXTUAL)	List of Things Important to the Business 	List of Processes the Business Performs 	List of Locations in which the Business Operates 	List of Organizations Important to the Business 	List of Events/Cycles Significant to the Business 	List of Business Goals/Strategies 	SCOPE (CONTEXTUAL)
<i>Planner</i>	ENTITY = Class of Business Thing	Process = Class of Business Process	Node = Major Business Location	People = Major Organization Unit	Time = Major Business Event/Cycle	Ends/Means = Major Business Goal/Strategy	<i>Planner</i>
BUSINESS MODEL (CONCEPTUAL)	e.g. Semantic Model  Ent = Business Entity Rel = Business Relationship	e.g. Business Process Model  Proc = Business Process IO = Business Resources	e.g. Business Logistics System  Node = Business Location Link = Business Linkage	e.g. Work Flow Model  People = Organization Unit Work = Work Product	e.g. Master Schedule  Time = Business Event Cycle = Business Cycle	e.g. Business Plan  End = Business Objective Means = Business Strategy	BUSINESS MODEL (CONCEPTUAL)
<i>Owner</i>							<i>Owner</i>
SYSTEM MODEL (LOGICAL)	e.g. Logical Data Model  Ent = Data Entity Rel = Data Relationship	e.g. Application Architecture  Proc = Application Function IO = User Views	e.g. Distributed System Architecture  Node = IS Function (Processor, Storage, etc.) Link = Line Characteristics	e.g. Human Interface Architecture  People = Role Work = Deliverable	e.g. Processing Structure  Time = System Event Cycle = Processing Cycle	e.g. Business Rule Model  End = Minimal Assumption Means = Action Association	SYSTEM MODEL (LOGICAL)
<i>Designer</i>							<i>Designer</i>
TECHNOLOGY MODEL (PHYSICAL)	e.g. Physical Data Model  Ent = Segment/Table/etc. Rel = Pointer/Key/etc.	e.g. System Design  Proc = Computer Function IO = Data Elements/Sets	e.g. Technology Architecture  Node = Hardware/Systems Software Link = Line Specifications	e.g. Presentation Architecture  People = User Work = Screen Format	e.g. Control Structure  Time = Execute Cycle = Component Cycle	e.g. Rule Design  End = Condition Means = Action	TECHNOLOGY MODEL (PHYSICAL)
<i>Builder</i>							<i>Builder</i>
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	e.g. Data Definition 	e.g. Program 	e.g. Network Architecture 	e.g. Security Architecture 	e.g. Timing Definition 	e.g. Rule Specification 	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
<i>Sub-Contractor</i>							<i>Sub-Contractor</i>
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

John A. Zachman, Zachman International

Kuva 8: Zachmanin viitekehys [Zachman, 2003]

4.2.2 RM-ODP

Reference Model of Open Distributed Processing (RM-ODP) on standardoitu oliokeskeinen viitemalli, joka on tarkoitettu erityisesti avoimen ja hajautetun tietojärjestelmän kehitykseen. Tämä tarkoittaa lyhyesti sitä, että tietojärjestelmän osat voivat olla heterogeenisiä sisäisiltä toteutuksiltaan ja sijaita hajautetusti eri organisaatioissa, mutta ne pystyvät toimimaan yhteen avoimien rajapintojen avulla.

Standardissa on neljä perusosaa: 1) yleiskuva, 2) perusta (foundation), 3) arkkitehtuuri ja 4) arkkitehtuurin semantiikka. Viitemallin sisältämät määrittelyt ja kansainväliset standardit määrittelevät käsitteet, joita tarvitaan tietojärjestelmän suunnittelussa RM-ODP:n viidestä eri näkökulmasta. ODP sisältää viitekehysten suurten, hajautettujen järjestelmien määrittelytyön rakenteellistamiseksi. [RM-ODP, 2009]

Järjestelmäkehityksen viitekehyksessä on neljä keskeistä osaa:

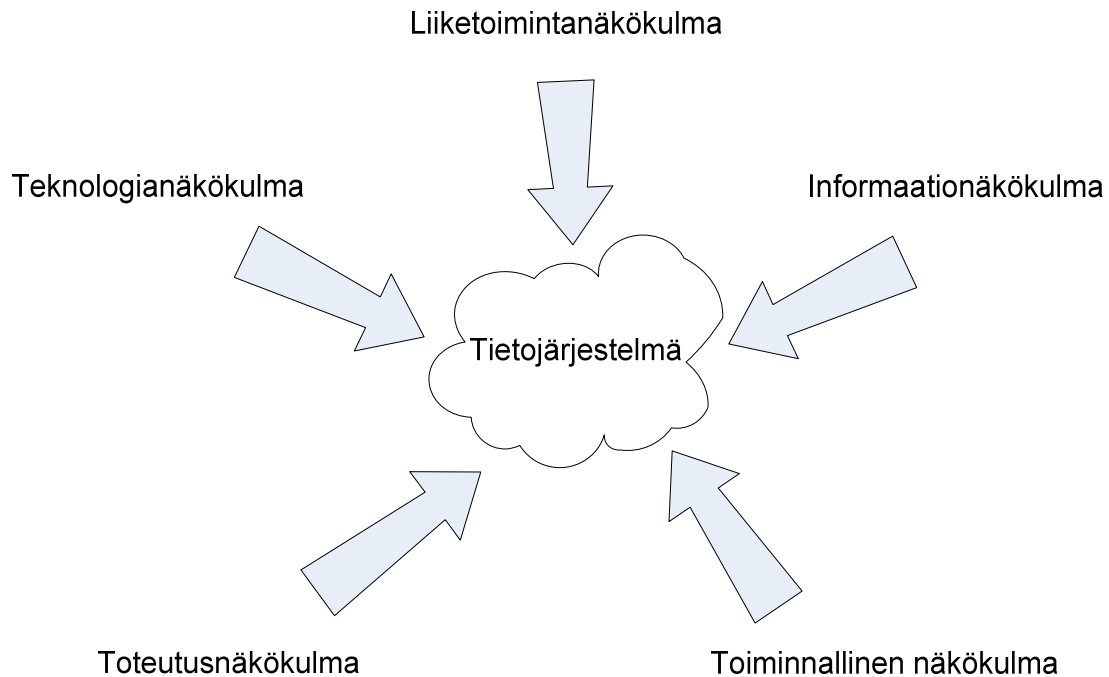
- Oliomallinnus (object modeling) järjestelmämäärittelyn lähestymistapana
- Järjestelmän määrittely erillisten, mutta toisiinsa liittyvien näkökulmien kautta
- Järjestelmän infrastruktuurin määrittely siten, että hajautus on läpinäkyvä järjestelmäsovelluksille
- Viitekehys järjestelmän yhdenmukaisuuden arvioimiseksi.

RM-ODP on vahvasti olioparadigmaan perustuva, mistä ajatellaan seuraavan monia hyötyjä, koska oliomallinnus mahdollistaa abstrahoinnin ja kapseloinnin. Järjestelmän toiminnallisuus on helposti erotettavissa toteutusratkaisusta ja kapseloinnin myötä heterogeenisyys voidaan kätkeä, virheet jäävät paikallisemmiksi ja palveluntarjoamisen mekanismit ovat palvelun käyttäjälle näkymättömiä.

RM-ODP:iin sisältyy viiteen näkökulmaan jakautuva viitekehys (Kuva 9). Mallissa näkökulmia ei haluta ajatella tasoina, jotka hierarkkisesti seuraisivat toisiaan. Vaikka käytännössä jotkin asiat tehdään ennen toisia, jokainen näkökulma on suorassa suhteessa muihin mallin näkökulmiin. Mallissa korostuu ajattelutapa, että kutakin näkökulmaa edustaa jokin kohderyhmä, jolla on oma kieli ja intressit. Siten jokainen näkökulma on riippuvainen toisesta. Näkökulmille on määritetty kieli tai terminologia, joka optimoi sanaston ja esitystavan kunkin näkökulman kohderyhmän mukaiseksi [RM-ODP, 2009; Tyndale-Biscoe, 2002]. RM-ODP:n näkökulmia ovat:

- Liiketoimintanäkökulma (enterprise viewpoint), kohdeorganisaation aktiviteetteja ohjaavat päämäärät, laajuus ja käytännöt; järjestelmän tarkoitus;
- Informaationäkökulma (information viewpoint); järjestelmän käsittelemät tiedot ja tiedon tulkintaan ja käyttöön liittyvät rajaukset; mistä järjestelmä koostuu;
- Toiminnallinen näkökulma (computational viewpoint), järjestelmän toiminnallinen hajauttaminen olioiksi, jotka keskustelevat rajapintojen kautta; miten järjestelmä toimii;

- Toteutusnäkökulma (engineering viewpoint), infrastruktuuri, joka tarvitaan järjestelmän levittämiseen; miten järjestelmä toimii;
- Teknologianäkökulma (technology viewpoint), tarvittavien teknologioiden valitseminen; keinot järjestelmän kokoamiseksi.

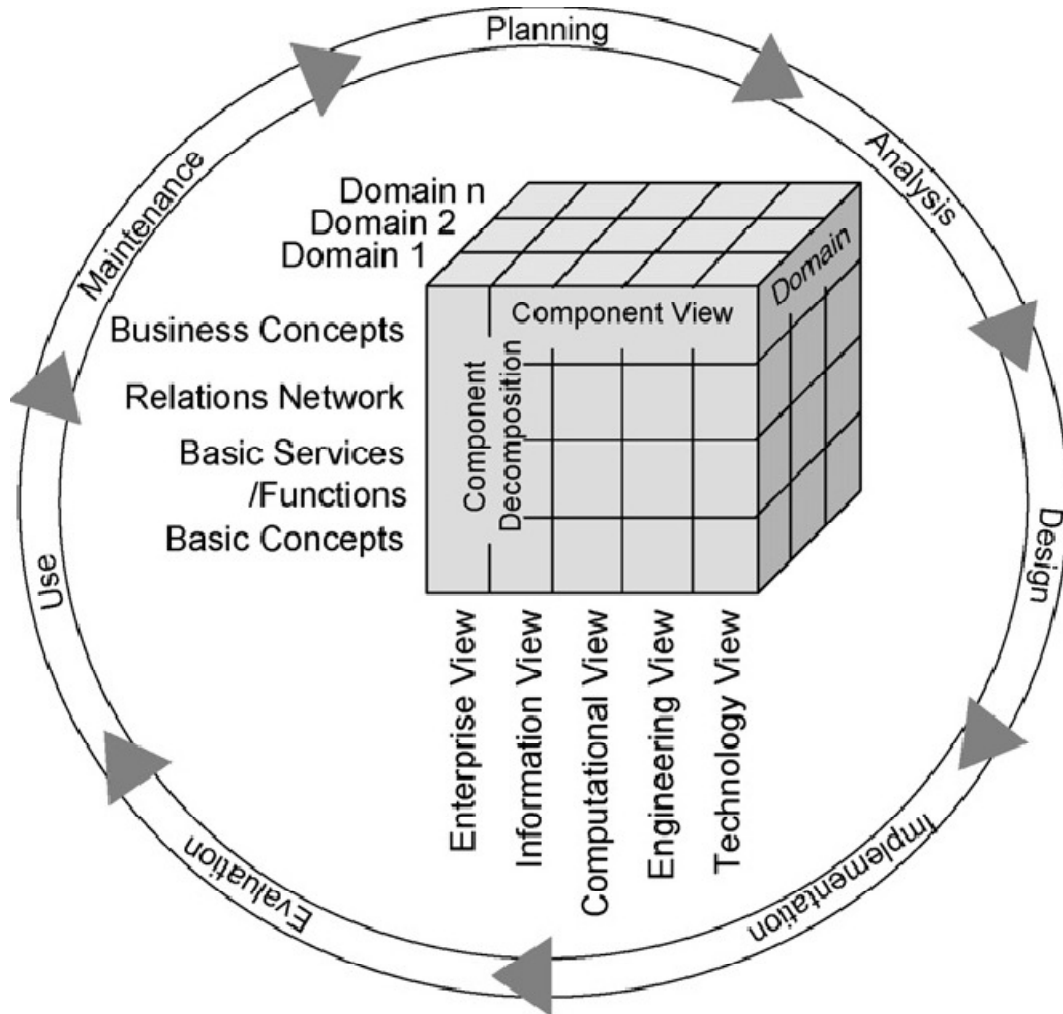


Kuva 9: RM-ODP näkökulmat [RM-ODP, 2009; Tyndale-Biscoe, 2002]

4.2.3 Generic Component Model

Generic Component Modelin (GCM) viitekehys on tarkoitettu kuvaamaan mikä tahansa todellisen maailman järjestelmä. Se kuvaa objektien sisäisten ja niiden välisten riippuvuuksien yhteiset toiminnallisuudet nojautuen komponenttiparadigmaan pelkän objektijattelun sijasta. Kokonaisuus hajautetaan komponenteiksi, esimerkiksi yrityksen eri toimintojen tai toimialojen (domain) mukaisiksi osakokonaisuuksiksi, kuten laskutus, markkinointi ja varasto. Komponentit ja niiden väliset yhteydet kuvataan ja lopuksi koostaan ne kohdealueen mukaiseksi kokonaisuudeksi. Mallissa vähennetään kohdejärjestelmän kompleksisuutta kolmiulotteisella mallilla (Kuva 10), jossa ensimmäinen ulottuvuus yksinkertaistaa analyysia erottelemalla todellisessa maailmassa toisiinsa kytkeytyvät sovellusalueet (domain) esimerkiksi laskutukseen, hallinnolliseen, tekniseen ja sosiaaliseen osaan. Toinen ulottuvuus purkaa rakenteellista kompleksisuutta. Siinä käsitteet ja rakenne puretaan karkealta tasolta yksityiskohtaiseksi liiketoimintakonsepteista suhdeverkostoihin, peruspalveluihin ja -toiminnallisuuksiin ja lopulta peruskäsitteisiin. Tämän jälkeen tuotetut järjestelmäkomponentit nivotaan takaisin yhteen mallin kolman-

nessa ulottuvuudessa, joka etenee viisivaiheisesti liiketoimintanäkökulmasta teknologianäkökulmaan. Viitekehyksen viisi näkökulmaa pohjautuvat RM-ODP:hen. Komponenttiajattelussa on hyvin paljon yhtäläisyyksiä SOA-ajatteluun (Service-Oriented Architecture), johon Suomen kansallinen arkkitehtuuri perustuu [STM, 2007a].



Kuva 10: Generic Component Model [Lopez and Blobel, 2009].

4.3 Tietoarkkitehtuuri

Tietoarkkitehtuuri (IA, Information Architecture) on korkean tason kuvaus yrityksen (kohdealueen) aktiviteeteista ja sen toiminnan edellyttämistä tiedoista. Sen tulisi olla mahdollisimman riippumaton organisaation rakenteesta, henkilöstöstä ja teknologisista ratkaisuista. Tietoarkkitehtuurin avulla saavutetaan ja ylläpidetään järjestelmien käsitteellistä ja rakenteellista yhteistoiminnallisuutta. Se voidaan muodostaa aktiviteettien pohjalta tuotetuista loogisista tietomalleista, esimerkiksi ER-malleista. [Cisneros et al.,

1997; Brancheau & Wetherbe, 1986] Konkreettisesti tietoarkkitehtuuri kertoo, mitä tietoa tallennetaan, missä muodossa kukin tietoelementti tallennetaan, mihin tieto tallennetaan ja missä sitä käytetään.

Kolme merkittävää syytä kasvaneeseen kiinnostukseen informaatio suunnittelua eli informaation rakenteen, käytön ja esittämisen suunnittelua kohtaan ovat Brancheaun ja Wetherben [1986] mukaan:

- 1) Informaation arvo ja voimakkaasti kasvavat kustannukset
- 2) Informaation käytön ja tallentamisen hajautuminen
- 3) Informaation lisääntyvä käyttö kilpailuetuna.

Yli kahdenkymmenen vuoden jälkeen nämä samat perusteet ovat edelleen vahvoilla jos eivät ole käyneet entistäkin tärkeämmiksi tietotekniikan tultua yhä isommaksi osaksi arkea. Ensinnäkin informaatio on arvokasta ja sen hallinta maksaa. Mitä huonommin tietoa hallitaan, sitä korkeammat ovat hallinnan kustannukset. Lisäksi huonon tiedonhallinnan seuraukset kumuloituvat. Samaan tapaan kuin yritystoiminnassa, myös terveydenhuollossa merkittävä osa työstä kaikilla organisaation tasoilla liittyy tiedon tallentamiseen, syntetisointiin ja levittämiseen. Toisekseen tietoa käytetään ja tallennetaan lukuisissa paikoissa, joten on tärkeätä suunnitella tiedonhallinta siten, että redundanttia ja ristiriitaista tietoa ei synny, koska siitä voi seurata virheitä ja ylimääräistä työtä. Kaikkien käytössä olevien järjestelmien on toimittava yhteen, samaa tietoa ei tallenneta kahden paikkaan. Kolmas kohta, informaation tuoma kilpailuetu, ei julkisella sektorilla ole ehkä yhtä selkeä käsite kuin yrityksissä, mutta sen voi hyvin ajatella olevan tehokkuusvaatimuksen täyttäminen; se, että voidaan tuottaa entistä laadukkaampaa ja turvallisempaa hoitoa ja ohjata säästyneitä varoja muihin toimintoihin, mistä seuraa yhteiskunnallista hyvinvointia.

Yrityksen toiminnassa *miten (how)* muuttuu paljon useammin kuin *mitä (what)*. Hyvin tehty tietoarkkitehtuuri vastaa kysymykseen mitä, eikä mieltä miten. Brancheau ja Wetherbe [1986] kertovat esimerkin autoteollisuudesta, jossa valmistusmenetelmät ovat kehittyneet valtavasti, mutta itse tuotteen käyttötarkoitukset ovat pitkälti samat kuin ennenkin. Muutoksia kuitenkin tulee ajan myötä, ja arkkitehtuuri on hyvin toteutettu, jos ne voidaan kytkeä mukaan joustavasti.

Cisneros ja muut [1997] ovat pohtineet perusteita tietoarkkitehtuurin tekemiselle:

- **Tiedon valtameri**

Samalla kun tiedon (datan) määrä jatkaa kasvamistaan, luotettavan ja täsmällisen tiedon saaminen käy hankalammaksi. Tiedonhankinnan kontrollointi auttaa, mutta sitä ei voi tehdä ellei instituutiolla ole käytössä ”katalogia” tiedosta.

- **Tiedon kahdentuminen**

Tiedon kahdentumisesta seuraa monia haittoja. Kun samat tiedot kerää useampi toimija, myös työmäärä kasvaa. Pällekkäisyys synnyttää ristiriitaista tietoa. Sillä on myös merkittäviä kustannusvaikutuksia.

- **Tiedon redundanssi**
Redundantti tieto on suoraa seurausta päällekkäisestä tietokantojen kehitystyöstä eri toimintayksiköissä. Redundanssi syntyy kun ei ole yhteistä tietoarkkitehtuuria joka orkestroisi kehitystyötä.
- **Yleisymmärtämys**
Tietoarkkitehtuuri antaa kohteestaan organisaation rakenteesta riippumattoman yleiskuvauksen, joka kestää infrastruktuurin muutokset tai muuttuu vain vähän aina niin kauan kuin toiminnan tarkoitus (autojen valmistaminen, sairauksien parantaminen) pysyy entisellään.
- **Päätöksenteon tuki**
Yritysten välisessä kilpailussa menestymiseen tarvitaan ajantasaista ja paikkansapitävää tietoa.
- **Raportoinnin lainsäädännön ja asetusten mukaisuus**
Esimerkiksi yliopistolle virheellisestä raportoinnista voi seurata rahoituksen pienentyminen tai rangaistustoimenpiteitä.
- **Tieto on tärkeä ja arvokas pääoma yritykselle**
Tietoarkkitehtuurin suunnittelu on edellisten perustelujen valossa hyvin kannattavaa, sillä se kasvattaa ja suojaa yrityksen pääomaa

Tietoarkkitehtuuri [Cisneros et al., 1997]

- helpottaa järjestelmien, prosessien, datan ja informaation integrointia,
- dokumentoi prosessit ja datan keskitetysti,
- tukee tiedonhallintaa,
- kasvattaa ymmärtämystä kohdealueesta ja edistää yhteistä sanastoa,
- tekee tiedosta yrityksen pääomaa,
- tunnistaa redundantin datan ja prosessit.

Tietoarkkitehtuurista puhutaan huomattavan usein myös esimerkiksi hakukoneiden ja www-sivujen suunnittelun yhteydessä. Tässä tutkimuksessa tarkoittamani tietoarkkitehtuuri ja tämä ehkä paremmin informaatiotutkimuksen alalle sijoittuva merkitys ovat hyvinkin sukua toisilleen, mutta jälkimmäisessä huomio kiinnittyy tietojärjestelmäteollisen lähestymistavan sijaan tiedon syntaksin, sanaston, rakenteen ja navigoinnin suunnitteluun esimerkiksi www-sivuja tai tiedonhakua varten, ja siihen yhdistyy tyypillisesti myös käytettävyyšnäkökulma. [Gilchrist and Mahon, 2004]

4.4 Arkkitehtuurin suunnittelumenetelmiä

Lopezin ja Blobelin [2009] mukaan arkkitehtuuriviitekehysten, kuten Zachmanin viitekehysten [1987; 2003], avulla voidaan kuvata laaja ja mutkikas kohdealue, mutta viitekehyksiin ei usein kuulu työkaluja tai prosessia tietojärjestelmän arkkitehtuurin ja sen osien yksityiskohtaisempaan suunnitteluun. Jotta kattava tietojärjestelmän kokonaisark-

kitehtuuri voitaisiin tuottaa, arkkitehtuuriviitekehystä on käytettävä yhdessä erilaisten viitemallien, kuten arkkitehtuurimallien ja tietoarkkitehtuurimallien, metodologioiden, sanastojen ja kuvaustapojen kanssa.

Kehitys- tai suunnittelumenetelmä (development framework) on systemaattinen tietojärjestelmän suunnitteluprosessin kuvaus, johon kuuluu kokoelma menetelmiä ja malleja, joiden avulla on mahdollista päästä haluttuun lopputulokseen. Ne voivat olla hyvin generisiä tai erityisesti tiettyyn kohdealueeseen tarkoitettuja. Esimerkiksi HDF-menetelmä (HL7 Development Framework) [SerAPI, 2006] kuvaa sovellusrajapintojen välille tarvittavan sanomavälityksen suunnitteluprosessin varsin yksityiskohtaisesti. Vaikka HDF-mallin prosessi on yksityiskohtainen ja lähtenyt terveydenhuollon tarpeista, se tukeutuu UML:ään ja on tarkoitettu muuhunkin kuin terveydenhuollon tarpeisiin. Suunnittelumenetelmää voi pitää sitä luotettavampana perustana tietojärjestelmän suunnitteluprosessille, mitä pidempään sitä on käytännössä ehditty koetellaan ja sen myötä kypsyttämään. Toinen valintaan vaikuttava peruste on suunnittelumenetelmän vahva teoriapohja.

Kohdealueen viitemallien myötä saadaan käyttöön jo aiemmin tuotettua kohdealueeseen soveltuvaa tietoa, varmistetaan tuotetun arkkitehtuurin kattavuus ja harmonisuus olemassa olevien ratkaisujen kanssa. Esimerkiksi Zachmanin malli ei määritä mitään tiettyjä kuvausmenetelmiä tai viitemalleja käytettäväksi; se antaa raamit, mitä kaikkia asioita kuvauksen tulee kattaa, mutta ei tarjoa konkreettisia työkaluja tai prosessia arkkitehtuurin eri vaiheiden toteuttamiseen. Se olisikin vaikeasti perusteltavissa, koska samalla viitekehysten yleiskäyttöisyys katoaisi.

Tietoarkkitehtuuri on rajapinta liiketoiminnan prosessien ja tietojärjestelmän toiminnallisuuden välillä ja siksi sitä suunniteltaessa tulee tietää, mihin tarkoitukseen ja millaiseen ympäristöön se sijoittuu. Mikäli olemassa olevissa järjestelmissä on aiemmin hyödynnetty tai sovellusalueelle on tarjolla viitearkkitehtuureja ja -malleja, niitä käyttämällä suunnittelijan työ voi helpottua huomattavasti. Ne auttavat varmistamaan, että tietovaatimukset harmonisoidaan tietoarkkitehtuurissa siten, että tietojärjestelmästä tulee yhtenäinen ja kattava sisäisesti ja yhteistoiminnallinen niiden järjestelmien kanssa, joihin se mahdollisesti liitetään. Tämä luonnollisesti edellyttää, että olemassa olevat järjestelmät on tehty samojen viitemallien säännösten mukaisesti. Epäilemättä viitemallit parantavat myös tietojärjestelmähankkeen onnistumisen mahdollisuuksia, koska suunnittelija voi perustaa työnsä aiemmin saatuun kokemukseen ja tietämykseen. Monet terveydenhuollon viitemalleista ja suunnittelumenetelmistä ovat standardoituja tai ovat valmistumassa standardeiksi [Blobel, 2007; CEN EN 12967-2:2007; SerAPI, 2006], mikä lisää luottamusta niiden teoriaperustaan.

Terveydenhuollon tietojärjestelmäkehitykseen on tarjolla useita arkkitehtuuristandardeja, tietoarkkitehtuuriviitemalleja, arkkitehtuurin kuvauskieliä ja sanastoja [mm. Blobel, 2007; Lopez and Blobel, 2009]. Eri osa-alueita varten on tarjolla yksityiskohtaisia kuvauksia tarvittavista toiminnoista ja tiedoista. Monen mallin kohteena on sähköi-

nen potilaskertomusjärjestelmä, joka onkin keskeinen elementti terveydenhuollon tiedonkäsittelyssä. Lopez ja Blobel [2009] toteavat, että moni standardi ei kuitenkaan kuvaa kattavasti sitä prosessia, jolla tietojärjestelmän arkkitehtuuri suunnitellaan.

Tietojärjestelmän suunnittelun eri vaiheissa voidaan hyödyntää eri lähestymistapoja ja eri tarkoituksiin suunniteltuja arkkitehtuureja voidaan yhdistellä. Valinnanvaraa on paljon eikä yhtä oikeaa polkua liene löytynyt, mutta tässä tutkimuksessa voidaan keskittyä tarkastelemaan sellaisia arkkitehtuureja, jotka kohdistuvat terveydenhuollon sektoreille, ovat laajasti tunnettuja ja joilla on keskeinen rooli tietoarkkitehtuurin suunnittelussa. Lopez ja Blobel [2009] ovat analysoineet eri arkkitehtuureja terveydenhuollon tietojärjestelmäarkkitehtuurin suunnittelumetodologian kehittämisen näkökulmasta ja jaotelleet sekä yleisiä että terveydenhuoltoon painottuvia arkkitehtuureja taulukon 2 mukaisesti.

Taulukko 2: Arkkitehtuurisia lähestymistapoja Lopez and Blobel [2009] mukaan.

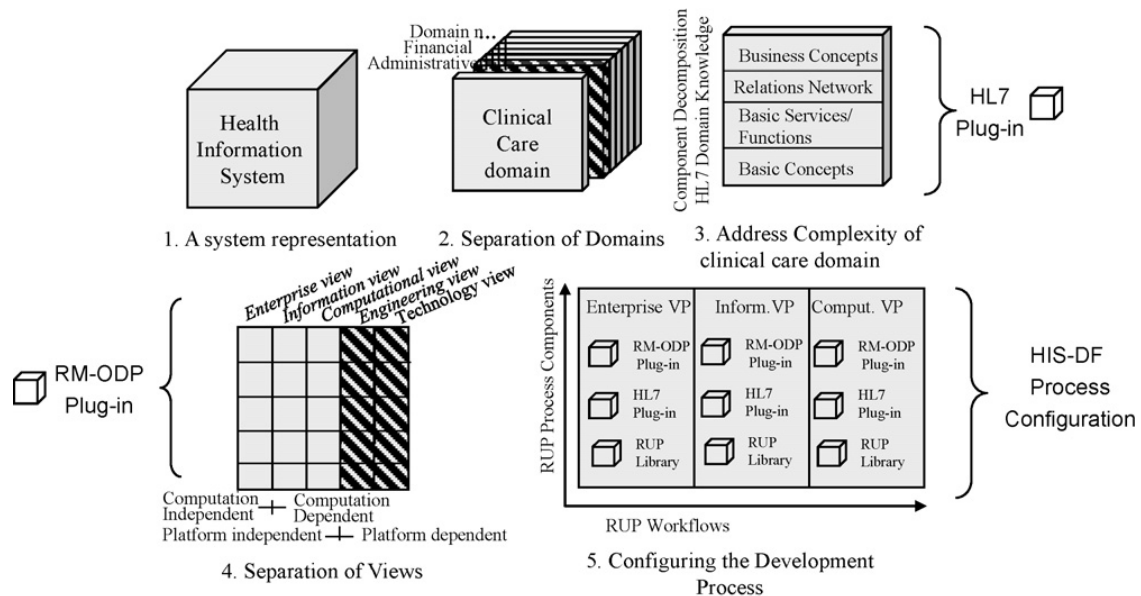
Arkkitehtuuriset lähestymistavat	Yhteenkuuluvat arkkitehtuurit	Kohdealue (Domain)	
		Ohjelmistotuotanto	Terveydenhuollon informatiikka
Arkkitehtuuriviitekehukset	Liiketoiminta-arkkitehtuurit	Zachman; FEAF; DODAF; RM-ODP; TOGAF; IEEE 1471-2000	Generic Component Model (GCM)
Arkkitehtuurimallit	Viitemallit; arkkitehtuurityylit	Objektiorientoituneet arkkitehtuurit; komponenttiperustaiset arkkitehtuurit; MDA; SOA; liiketoimintaprosessimallit	ISO EN 13606-1; OpenEHR; HL7 RIM; CDA; GCM specialisation
Kuvauskielet	Mallinnuskielet	UML; ACME ADL; OCL	OpenEHR ADL
Middleware-arkkitehtuurit	Komponenttiarkkitehtuurit; Palveluarkkitehtuurit	Web services; CORBA; J2EE; COM; .NET	OMG HDTF; CEN 12967
Arkkitehtuurin kehittämisprosessi	Analyysi/Suunnittelumetodit ja mallit	RUP; SAAM; ATAM; ADD; TOGAF ADM; SEI CMMI; SPEM	HL7 HDF; HL74SOA; CEN EN 12967-1

Seuraavassa esittelen joitakin terveydenhuollon tietojärjestelmien suunnittelumenetelmiä ja viitemalleja, joita voisi mahdollisesti soveltaa myös lääkitystiedon tietoarkkitehtuurin toteuttamisessa.

4.4.1 HIS-DF

HIS-DF (development framework for semantically interoperable health information systems and components) on taulukossa 2 esitellyistä keskeisimpien terveydenhuollon viitemallien ja menetelmien arvioinnin sekä harmonisoinnin tulos. Se on menetelmä semanttisesti yhteistoiminnallisten terveydenhuollon tietojärjestelmien ja komponenttien kehittämiseen. HIS-DF on laaja-alainen arkkitehtuurin suunnittelumenetelmä, joka pyrkii ylittämään perustanaan olevien menetelmien rajoitteet. Se kattaa liiketoimintanalyysin, vaatimusmäärittelyn ja suunnittelun sekä kaikkien vaiheiden dokumentaation. Malli on alustariippumaton ja jättäytyy GCM:n kolmannelle tasolle, toiminnallisuusnäkökulmaan. Se ei siten sisällä suunnittelumenetelmiä toteutus- ja teknologianäkökulmalle.

HIS-DF samanaikaisesti laajentaa GCM-viitekehystä ja systemaattisesti rajaa GCM:n fokusta kohdealueeseen viidessä vaiheessa (Kuva 11).



Kuva 11: HIS-DF ydinprosessi [Lopez and Blobel, 2009; HIS-DF]

Kuvassa 11 esitetty HIS-suunnittelumenetelmä etenee viisivaiheisesti [Lopez and Blobel, 2009; HIS-DF]:

- 1) Koska GCM soveltuisi kuvaamaan mitä tahansa järjestelmää, HIS-DF:n kohdealueeksi rajataan terveydenhuollon tietojärjestelmät
- 2) Eri sovellusalueista (domain) erotellaan kohteeksi kliininen hoitotyö. Pilkkomisen tarkoituksena on vähentää analyysin kompleksisuutta, myöhemmin sovellusalueiden yhteiset toiminnot ja tiedot liitetään analysoimalla niiden väliset yhteydet
- 3) Kolmannen vaiheen tarkoitus on keskittyä terveydenhuollon tietojärjestelmän kompleksisuuteen purkamalla se karkealta tasolta liiketoiminnan käsitteistä aina peruskäsitteisiin. Tässä vaiheessa määritellään osa-alueen (domain) sisältämä tieto – liiketoiminnan viitemallit, informaatioviitemallit, sovellusalueen käsitteet ja sanasto, joita tarvitaan semanttiseen integraatioon. Näiden tietojen määrittelyssä hyviä tiedonlähteitä sekä vertailukohtia ovat esimerkiksi HL7 RIM, kohdealueen informaatiomallit (domain information models), CDA, toiminnalliset palvelumäärittelykset, OpenEHR-arkkityypit ja terveydenhuollon tietojärjestelmien arkkitehtuuristandardit (CEN EN 12967, OMG HDTF)
- 4) Edellisissä vaiheissa suoritetun liiketoiminta-arkkitehtuurin analyysin jälkeen määritellään suunniteltavan arkkitehtuurin laajuus rajaamalla se alustariippumattomiin osiin RM-ODP-mallista
- 5) Koska RM-ODP-standardiin ei kuulu tiettyä notaatiota tai tapaa eri näkökulmien mallien kuvaamiseen, liitetään mukaan RUP:iin (Rational Unified Process) perustuvat komponentit tukemaan kunkin näkökulman toteuttamista. Niitä ovat roolit, tuotteet (work products), artifaktat, ohjeet, vaiheet, iteraatiot ja työvirrät.

4.4.2 ISO/TC 215

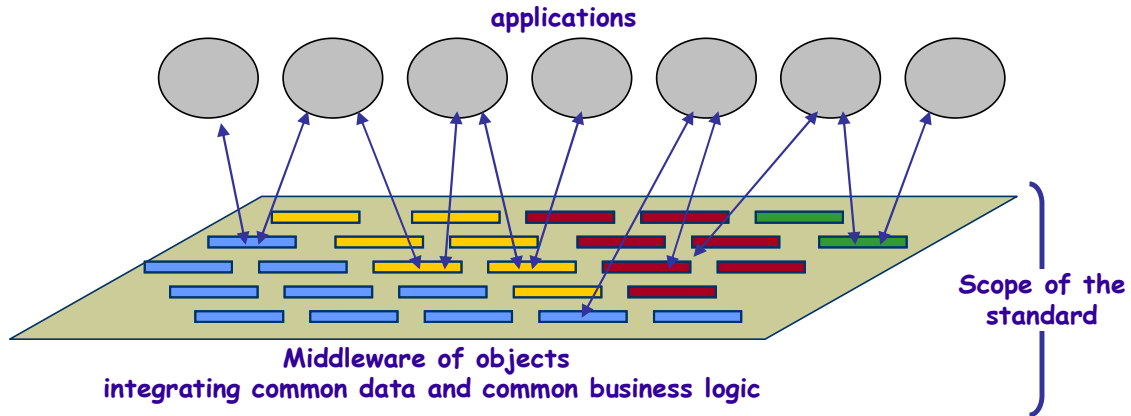
HISA (Health Informatics – Service architecture) [CEN EN 12967-2:2007] on palvelu-arkkitehtuuristandardi, joka kuvaa terveydenhuollon tietojärjestelmän middlewaren eli palvelurajapinnan (Kuva 12), joka integroi ohjelmistokomponentit ja palvelut. Standardi koostuu kolmesta osasta:

- 1) liiketoimintanäkökulma (Enterprise viewpoint)
- 2) informaationäkökulma (Information viewpoint)
- 3) toiminnallinen näkökulma (Computational viewpoint).

Standardi pohjautuu RM-ODP:iin, mutta keskittyy kolmeen näkökulmaan jättäen toteutus- ja teknologianäkökulmat standardin ulkopuolelle. Standardi pyrkiikin tietoisesti irrottautumaan toteutustavoista ja teknologiaratkaisuista.

Standardin toinen osa, informaationäkökulma, on tarkoitettu opastamaan tietojärjestelmän kuvaamisessa, suunnittelussa ja kehittämisessä tukien myös olemassa olevien järjestelmien integraatiota sekä organisaation sisäisesti että ulkoisesti. Informaationäkökulma määrittelee semantiikan tietomallille, jonka avulla integroidaan liiketoiminnan tiedot ja tietotarpeet, jotka on formalisoitu edeltävällä liiketoimintatasolla. Informaa-

tionäkökulma on taso, jolla palvelurajapinta yhdistää tiedon ja liiketoimintalogiikan. Standardimallia voi myös laajentaa kohdealueen tarpeiden mukaisesti.



Kuva 12: HISA-standardin kohdealue [CEN EN 12967-2:2007]

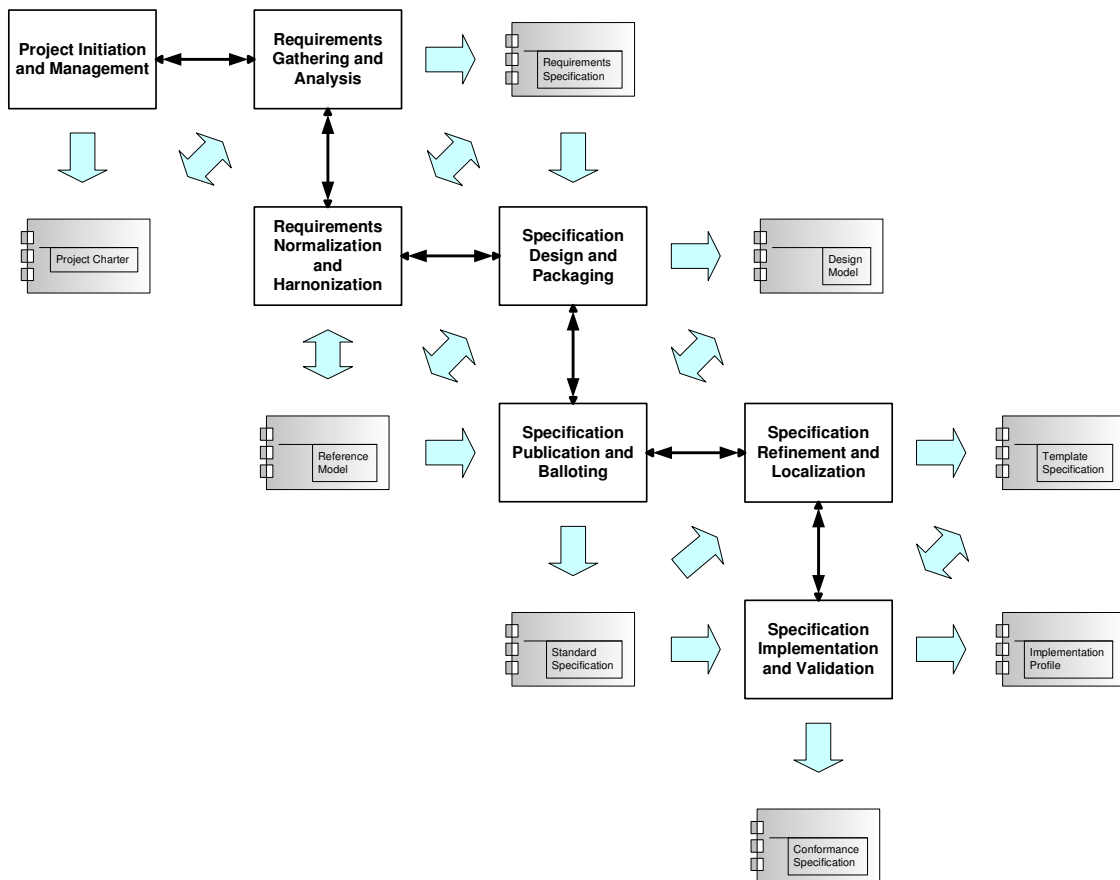
4.4.3 HL7 Development Framework (HDF)

HL7 (Health Level 7) on terveydenhuollon sovellusalueen voittoa tavoittelematon standardointiorganisaatio, joka tuottaa ohjelmien yhteistoiminnallisuutta tukevia standardeja. Se ei painotu mihinkään tiettyyn sovellusalueeseen vaan fokuksessa on terveydenhuollon sovellusalueiden rajapintojen vaatimukset. Tavoitteena on kehittää johdonmukaisia laajennettavissa olevia standardeja, jotka mahdollistavat rakenteellisen tiedon välittämisen sanomina eri ohjelmien tai järjestelmien välillä. Toinen keskeinen tavoite on formaali metodologia tukemaan HL7-standardien johtamista HL7 RIM -mallista. [HL7, 2009a]

HL7 Development framework (HDF) on edeltäjästään HL7 Message Development Frameworkista (MDF) laajennettu mallipohjainen viitekehys. MDF on tarkoitettu erityisesti HL7-sanomien määrittelyyn. HDF eroaa sovellusalueen kattavuudessa, joka laajentaa malliperustaisen prosessin muihinkin teknisiin määrittelyihin sanomanvälityksen lisäksi, on yhdenmukainen UML:n kanssa ja sisältää menettelyt ylläpitoon ja versiointiin. [SerAPI, 2006] HDF-metamallissa on pyritty harmonisoimaan MDF- ja UML-metamallit.

HDF-metodologian kehitysprosessi on seitsemänvaiheinen. Jokainen prosessin vaihe on riippuvainen edeltävän vaiheen tuottamasta dokumentaatiosta, mutta prosessi on iteratiivinen siten, että edelliseen vaiheeseen voidaan palata toisin kuin vesiputousmallissa. [Shakir, 2003; Serapi, 2006] Kuvassa 13 on esitetty sekä kehitysprosessin vaiheet että niihin liittyvä dokumentaatio. Kehitysprosessin vaiheet ovat:

1. Projektin aloittaminen (Project initiation)
2. Vaatimusten dokumentointi (Requirements documentation)
3. Määrittelyn mallintaminen (Specification modeling)
4. Määrittelyn dokumentointi (Specification documentation)
5. Määrittelyn hyväksyminen (Specification approval)
6. Määrittelyn julkaiseminen (Specification publication)
7. Toteutuksen profilointi (Implementation profiling).



Kuva 13: HL7-kehitysmenetelmä [Shakir, 2003]

Kunkin kehitysprosessin vaiheen syötteet, toiminnot, työkalut, osallistujat ja tulokset (output) on kuvattu yksityiskohtaisesti HDF:n dokumentaatioissa [HDF, 2005], mutta tässä yhteydessä on aiheellista keskittyä vain yleispiirteisiin sekä tarkemmin vaatimusmäärittelyvaiheeseen ja sen harmonisointiin liittyvään osioon.

HDF-prosessi tukeutuu vahvasti UML:ään ja prosessin eri vaiheissa tuotetut kuvat ja kaaviot ovatkin UML-muotoisia kaikilta niiltä osin kuin on sovellettavissa.

Projektin toisen vaiheen tuotoksena on sovellusalueen vaatimusmäärittely. Sen lopussa tuotettu malli harmonisoidaan HL7 viitemallien kanssa [SerAPI, 2006; HDF,

2005]. HDF-taustaselvityksen [SerAPI, 2006, s. 16] mukaan vaatimusmäärittelyvaiheen dokumentaation tuloksena on:

- 1) ”Toimintaprosessin dynaamisen käyttäytymisen ja rakenteen kuvaaminen storyboardien, liittymäkaavioiden ja yhteistyökaavioiden avulla
- 2) Aktiviteettikaavioiden avulla kuvatut terveydenhuollon toimintaprosessit, joita tarvitaan tiedonvaihdossa
- 3) Sovellusalueen analyysimalli – staattinen kuvaus UML-luokkakaaviolla, jossa ovat mukana toimintaprosessiin liittyvät käsitteet (mukaan lukien tiedonvaihdossa tarvittavien tietojen rakenne ja suhteet)
- 4) Sovellusalueen käsitteistö, jossa on huolellisesti ja täydellisesti määritelty jokainen sovellusalueen analyysimallissa esitetty käsite, käsitteeseen liittyvät ominaisuudet
- 5) Toimintasäännöt, joita voi pitää käsitteistön laajenuksena. Niiden avulla kuvataan tarvittavat rajoitteet vastaamaan tiedon eheyttä, tiedon todennettavuutta tai toimintasääntöjä ympäristössä, jossa tietoa vaihdetaan
- 6) Harmonisoitu tai hyväksytty vaatimusmäärittely.”

Vaatimusten dokumentaation vaiheet ovat seuraavat [SerAPI, 2006, s. 16]:

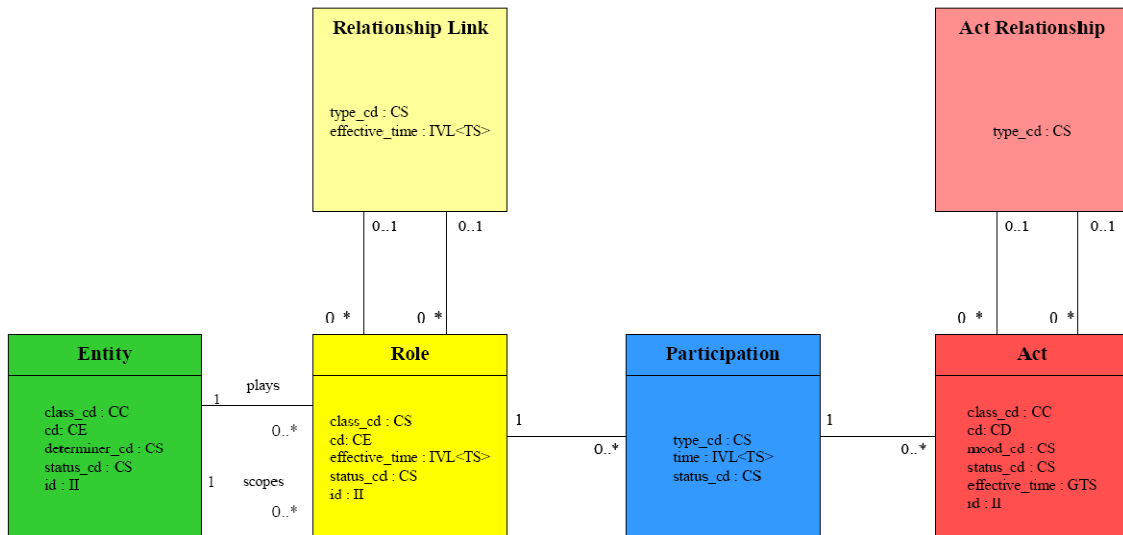
- 1) ”Toimintaprosessin dokumentointi: dynaaminen käyttäytyminen ja staattinen rakenne
- 2) Prosessin etenemisen dokumentointi: UML aktiviteettikaavio
- 3) Rakenteen dokumentointi: sovellusalueen analyysimalli (Domain Analysis Model) ja sanasto
- 4) Toimintasääntöjen dokumentointi: suhteet (Relationships), laukaisevat tapahtumat (Triggers) ja rajoitteet
- 5) Harmonisoidaan sovellusalueen analyysimalli (Domain Analysis Model) HL7:n viitemalleihin.”

Tietoarkkitehtuuria suunniteltaessa mielenkiinto kohdistuu erityisesti viimeiseen kohtaan, jossa tietovaatimukset harmonisoidaan HL7-viitemalleihin.

4.5 Terveydenhuollon tietoarkkitehtuuriviitemallit

HL7 RIM

HL7 RIM (Health Level 7 Reference Information model) on terveydenhuollon tarvitsemat tiedot ja niiden väliset suhteet kuvaava perustietomalli. Siitä johdetaan kaikki muut tietomallit ja -rakenteet [SerAPI, 2006] eli tarkennetut luokat jotka määrittelevät välitettävät sanomat. RIM on keskeinen osa sekä HDF-menetelmää että sitä edeltänyttä MDF:iä.



Kuva 14: HL7 RIM perusluokat [SerAPI, 2006]

Kuvassa 14 on esitelty RIM-mallin perusluokat, kokonainen viitemalli on liitteessä 1. Kullakin perusluokalla on oma väri, jonka perii myös aliluokka. Tässä tutkimuksessa käytetään SerAPI-taustaselvityksen [2006] suomennoksia, mutta täysin vakiintuneita suomenkielisiä vastineita käsitteille ei ole.

Toiminto (Act) tarkoittaa jotakin terveydenhuoltoon liittyvää, HL7:n kohdealueen, tapahtumaa. Toiminnon ilmentymä on tapahtumasta tehty merkintä.

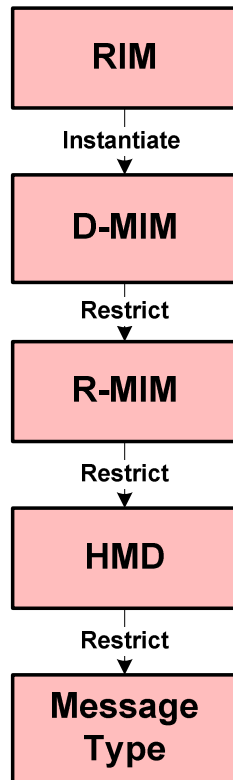
Entiteetti (entity) on jokin fyysisesti olemassa oleva asia, organisaatio tai asioiden joukko. Entiteeteiksi ei lasketa tietorakenteita, sähköisiä potilastietoja, viestejä jne.

Entiteetillä on **rooli** (role), jossa voi toimia ihminen tai esine/asia (thing). Ihmisen rooli voi olla asema, työ tai tehtävä, jossa hän sillä hetkellä toimii. Esine voi olla esimerkiksi jokin elintoimintoja seuraava laite. Tyypillisesti toisen entiteetin rooli on suorittaja ja toinen entiteetti on suorituksen kohde – lääkäri ja potilas.

Osallisuus (Participation) ilmaisee roolien ja toimintojen väliset yhteydet. Toimintoon on aina osallisena vähintään yksi rooli.

Roolien väliset linkit esitetään **Relationship Link** -luokalla ja **Act Relationship** -luokka kuvaa yhteyden kahden toimintoluokan välillä. [HL7 V3, 2009].

Kuvassa 15 on esitetty periaate, jolla HL7 RIM -mallista johdetaan tarkennetut mallit. D-MIM (Domain Message Information Model) on RIM-mallista johdettu kohde- tai sovellusalueen viestimalli. R-MIM (Refined Message Information Model) on sen tarkennettu tietomalli, jonka perusteella voidaan määrittellä varsinaisia viestejä, joita järjestelmät sitten välittävät. HMD (Hierarchical Message Description) on hierarkkinen tarkennettu viestin kuvaus. [SerAPI, 2006] Malleilla pystytään siis kuvaamaan kohdealueen tietoarkkitehtuuri ja etenemään yksityiskohtiin aina yksittäisiin viesteihin saakka.



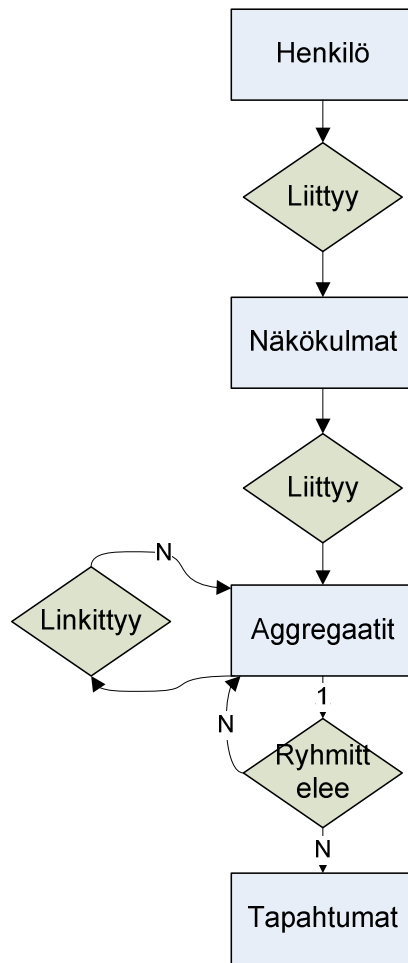
Kuva 15: HL7 RIM tarkennetut mallit [Shakir, 2003]

CMET (Common Message Element Types) on erityinen R-MIM:n muoto. CMET:it ovat eri sovellusalueiden mallien yhteisiä ja uudelleenkäytettäviä perusosia, tietokokonaisuuksia, joita voi käyttää HL7 v3 -sanomia rakennettaessa ja paikallistettaessa. Niitä voi siis käyttää sekä D-MIM:n että R-MIM:n osina. CMET:iä ei kuitenkaan ole tarkoitettu piilottamaan rakenteita kaavion yksinkertaistamiseksi esimerkiksi D-MIM:ssä, vaan CMET:in tuottamisen tarkoituksena on oltava uudelleenkäytettävyys. [HL7 V3, 2009]

4.6 Tietoarkkitehtuuri tiedon esittämisen ja käytön näkökulmasta

Plaisant ja muut [1998] kehittivät erilaisten henkilön historiaan liittyvien tapahtumien visualisoimista varten tietoarkkitehtuurin, jossa tapahtumia esitetään aikajanalla. Tutkimuksessa heillä on yhtenä esimerkkinä teorian soveltamisesta lääkityshistoria. Saman lääkkeen määräykset voidaan koota yhdeksi tapahtumaksi. Abstraktiotasoa nostettaessa voidaan samaan tarkoitukseen määrätyistä lääkkeistä, kuten beetasalpaajiin kuuluvista Propranololista ja Atenololista, koostaa oma tapahtumansa. Myös diagnoosi voisi olla tällainen kokoava yhtenä tapahtumana tarkasteltava kokonaisuus, johon voi kuulua useita lääkkeitä.

Jotta (lääkitys)tiedosta voitaisiin koostaa luontevasti erilaisia tarpeen mukaisia kokonaisuuksia, tarvitsee eri tietoelementtien välille luoda yhteyksiä (Kuva 16). Plaisantin ja muiden [1998] rakentamassa mallissa tietoarkkitehtuuriin kuuluu tapahtumia, näkökulmia (facet) ja aggregaatteja eli yhdistelmiä tai kokonaisuuksia sekä ristiviittauksia. Näkökulman tarkoitus on jaotella tapahtumia kategorioihin. Niitä voivat voi olla esimerkiksi lääkärikäynti, lääkitys tai laboratoriotulos, ja niillä voi myös olla hierarkia. Lääkitystiedon tapauksessa näkökulmia voisivat muun muassa olla ydintietomäärittelyjen mukaiset neljä kategoriaa: pitkäaikaislääkitys, tarvittaessa otettava pitkäaikaislääkitys, tilapäislääkitys ja lopetettu lääkitys. Hierarkkisesti nämä ovat lääkitystietonäkökulman alaisia. Aggregaatin idea on, että siihen kytkeytyvät tapahtumat todennäköisesti näytetään yhtenä kokonaisuutena. Se on toisiinsa vahvasti liittyvien ja samaan näkökulmaan kuuluvien tapahtumien ajallisesti rajoitettu kokonaisuus, esimerkiksi yhtä diagnoosia koskeva lääkitys voidaan koota yhdeksi tapahtumaksi. Aggregaateista voidaan muodostaa uusi korkeamman tarkastelutason tapahtuma. Ristiviittauksia käytetään kuvaamaan vähemmän vahvoja kytkeitä aggregaattien tapahtumien ja aikajaksojen välillä. Eri näkökulmiin kuuluvien tapahtumien välille voidaan tällä tavoin luoda yhteys, näin voitaisiin vaikkapa ilmaista tilanne, jossa samaa lääkettä on syöty kahteen eri tarkoitukseen.



Kuva 16: Tapahtumat, näkökulmat, aggregaatit ja ristiviittaukset [Plaisant et al., 1998]

Tutkimukseni tarkoituksena ei ole suunnitella visualisointia lääkitystiedolle, mutta Plaisantin ja muiden [1998] mallin ajatus tiedon kuvaamisesta erilaisina ajallisina jatkumoina ja tarkastelu eri laajuisina kokonaisuuksina on mielestäni huomioimisen arvoisen visio myös sille, mitä tietoarkkitehtuuria seuraavan prosessin lopputuloksena voitaisiin toteuttaa. Lääkitystiedon tietoarkkitehtuurin tulisi olla suunniteltu siten, että tiedon yhdisteleminen ja esittäminen edellä kuvatun kaltaisella tavalla olisi mahdollista.

4.7 Tietoarkkitehtuurin kehittämisprosessi

Analysoitaessa edellä esiteltyjä arkkitehtuuriviitekehyksiä, kokonaisarkkitehtuurin käsitettä ja arkkitehtuurin suunnittelumenetelmiä sekä arkkitehtuuriviitemalleja voidaan havaita, että niistä löytyy melko yhtenäiset suuntaviivat tietoarkkitehtuurin suunnittelu-prosessille. Tässä luvussa esitetty syntetisoitu kehittämisprosessi perustuu erityisesti HIS-DF, HDF- ja EN 12967 -malleihin. Tietoarkkitehtuuri johdetaan niissä rajaamalla kohdealue, kuvaamalla kohdealueen (liiketoiminnan) prosessi- ja tietovirrat tai käyttö-

tapaukset. Niistä syntyvä tietomalli harmonisoidaan kohdealueen viitemalleihin. Tietoarkkitehtuuri on esitellyissä malleissa poikkeuksetta se rajapinta, joka yhdistää liiketoiminnan prosessit tietojärjestelmän toiminnallisuuteen.

Kuvassa 17 on koostettuna geneerinen kuvaus tietoarkkitehtuurin suunnitteluprosessista, jonka tarkoitus on pääpiirteissään kuvata minkä tahansa tietoarkkitehtuurin kehittämiseen liittyvät vaiheet yleisellä tasolla. Tarkat toteutusmenetelmät ovat aina riippuvaisia kohdealueen luonteesta ja tietojärjestelmähankkeelle asetetuista vaatimuksista. Suunnitteluprosessin kuvauksessa mainitut ER, UML-dokumentit ynnä muut ovat tarkoitettuja esimerkeiksi mahdollisista vaihtoehdoista, ei suosituksiksi tai että ne kaikki pitäisi tuottaa. Myöskään GCM:n käyttö arkkitehtuuriviitekehyksenä ei ole edellytys, mutta tässä se on valittu ohjaamaan ajattelua komponenttikeskeisyytensä ja RM-ODP:stä periytyvien laajasti kohdealuetta tarkastelevien näkökulmien vuoksi.

Tietojärjestelmän suunnitteluprosessista voidaan erottaa neljä tietoarkkitehtuurin suunnitteluun liittyvää vaihetta (Kuva 17). Vaikkakin vaiheet on seuraavassa eritelty toisistaan, ne käytännössä ovat osin limittäisiä ja iteratiivisia.

1. Kohdealueen tutkimus ja rajaaminen

Kuvataan, millainen on kohdealue, mitä ovat liiketoiminnan päämäärät (objectives), mikä on tuotettavan järjestelmän laajuus ja käyttötarkoitus. Kartoitetaan sisäiset ja ulkoiset osapuolet (actors), joiden kanssa liiketoiminnan prosessit ovat kytköksissä. Kohdealue jaetaan komponenteiksi tai sovellusalueiksi, joista mallinnuksen kohteena saattaa olla yksittäinen tai useampi järjestelmän osa. Suunnitteluprosessin vaiheissa 2 ja 3 mallinnetaan komponentit yksi kerrallaan ja tunnistetaan niiden väliset suhteet.

Tulokseksi saadaan rajattu kohdealue ja liiketoimintaprosessin kuvaus.

2. Prosessi- ja tietovirtamallinnus, vaatimusmäärittelyt

Mallinnetaan kohdealueen toiminta prosessi-, tietovirta- ja käyttötapauksin. Tutkitaan, mitä tietoa tarvitaan, mikä tieto on liiketoiminnan kannalta keskeistä ja mitä ongelmia sen käyttöön liittyy.

Tuloksena saadaan vaatimusmäärittely, toiminnallinen kuvaus, storyboard, aktiviteettikaavio, käyttötapaukset sekä prosessi- ja tietovirtamallit.

3. Tietomallinnus

Edellisen vaiheen dokumentaatiosta tunnistetaan ja mallinnetaan kohdealueen tarvitsemat tiedot ja tiedon rakenne. Laaditaan lista käsitteistä ja niiden selityksistä.

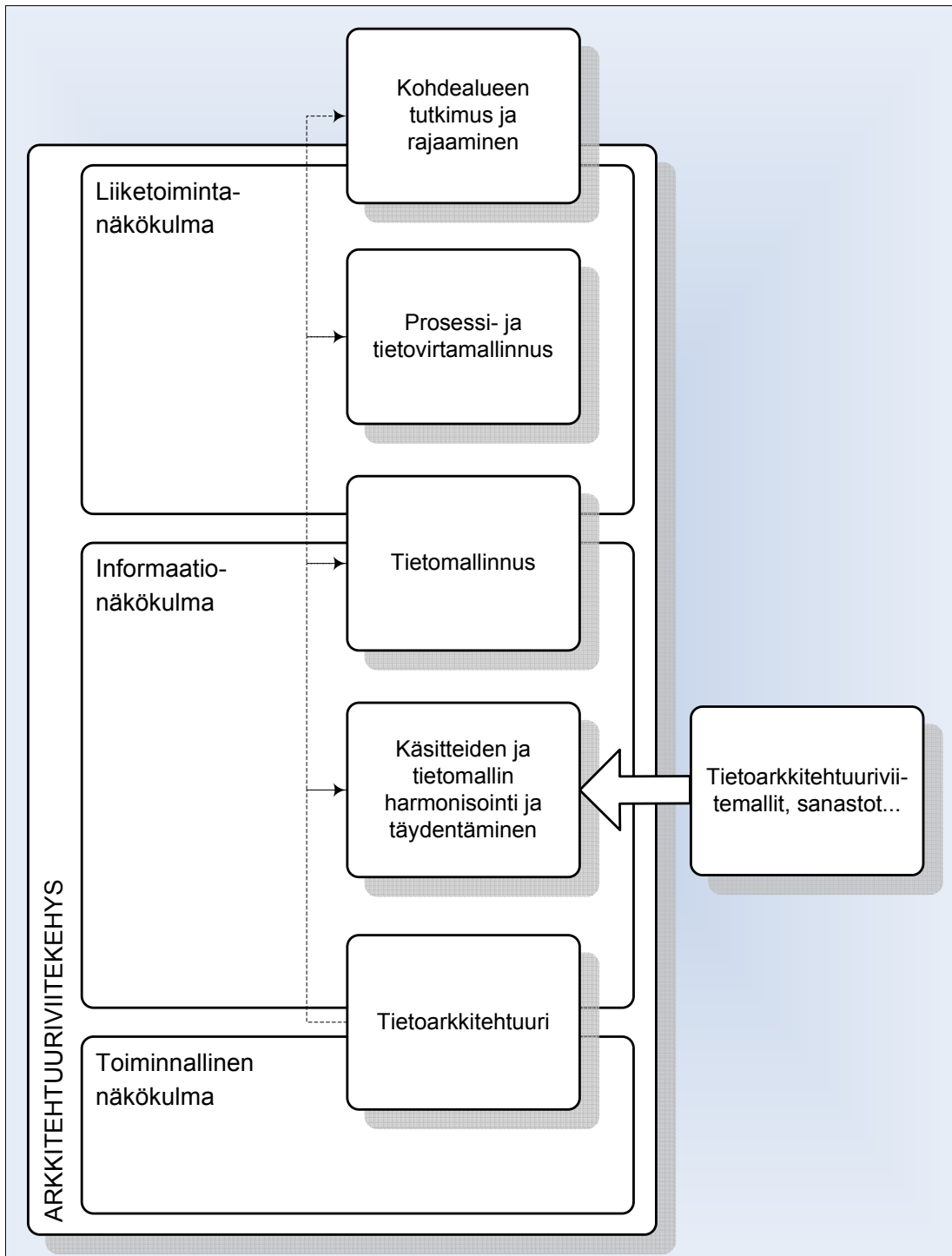
Tietomalli kootaan perustuen prosessien käyttämiin tai tallentamiin tietoihin sekä tunnistettuihin tietotarpeisiin, joita nykyiset prosessit eivät vielä tue. Prosessimallista ja tietovirroista voidaan löytää kohteet ja nii-

den ominaisuudet. Mallia täydennetään asiantuntijäläpikäynnein, jolloin voidaan täsmentää käsitteitä ja poistaa mahdollisia päällekkäisyyksiä. Tuloksena saadaan Domain analysis model, ER-malli ja sovellusalueen sanasto.

4. Käsitteiden ja tietomallin harmonisointi ja täydentäminen

Harmonisointi on prosessi, jossa tuotettua tietomallia verrataan valittuihin tietoarkkitehtuuriviitemalleihin, luokituksiin, käsitteistöön, sanastoihin ja määrittäisiin ja sovitetaan se niihin. Harmonisoinnin menetelmä ja välineet riippuvat valituista viitemalleista.

Lopputuloksena on kuvaus kohdealueen tietoarkkitehtuurista.



Kuva 17: Tietoarkkitehtuurin suunnitteluprosessi

5 Lääkitystiedon tietoarkkitehtuuri

Lääkitystiedon tietoarkkitehtuuri tulee suunnitella siten, että se vastaa lääkitystiedon käyttäjien tarpeita, lainsäädännön tarpeita ja on yhdenmukainen niiden standardien ja käsitteistöjen kanssa, joita käytössä olevat tai lähitulevaisuudessa käyttöönotettavaksi suunnitellut järjestelmät hyödyntävät. Suomessa on tapahtumassa lähivuosina kansallisen arkiston käyttöönotto, johon vähitellen kaikkien terveydenhuollon toimijoiden on liityttävä. Koska kansalliset suunnitelmat nojautuvat lokalisoituihin HL7-määrittelyihin, eikä muita yksityiskohtaisia määrittelyjä ole yhtä laajasti tarjolla, on luonteva ratkaisu harmonisoida lääkitystiedon tietoarkkitehtuuri HL7-määrittelyjen mukaisesti. HL7-määrittelyjen ytimessä on HL7 RIM -viitemalli. Sen käyttöä puoltavat kansallisen trendin lisäksi monet muutkin seikat:

- Viitemallin perusrakenne, jonka avulla on mahdollista kuvata periaatteessa mikä tahansa terveydenhuollon tapahtuma ja siihen osallistuvat ihmiset ja asiat
- Viitemallin fokus on terveydenhuollossa
- Attribuuttien periytyminen ja luokkien kloonaus
- Mahdollisuus yhdistää rooleja Relationship Link -luokalla ja toimintoja Act Relationship -luokalla silloinkin, kun yhteyttä ei synny periytyvyyden tai muun relaation kautta
- Viitemallista johdetut tarkennetut mallit tarjoavat johdonmukaisen tien sovellusalueen määrittelyyn tarkentamiselle aina toteutustasolle asti
- Runsaasti valmiita määrittelyjä eri sovellusalueille ja tiettyjen terveydenhuollossa säännöllisesti toistuvien tapahtumien tai asioiden ilmaisemiseen.
- Kattava valikoima tietotyyppisiä ja luokituksia
- Laajalti käytetty myös muualla maailmassa.

Tässä luvussa tarkastellaan, miten lääkitystiedon tietoarkkitehtuuri on muodostettavissa HL7 RIM -viitemalliin perustuen ja toteutetaan harmonisointi HL7 RIM -perusluokkien tasolla. Tietoarkkitehtuuri pohjautuu aiemmin tuotettuihin lääkitystiedon prosessi- ja tietomalleihin, kansalliseen viitekehukseen ja aiemmissa luvuissa käsiteltyihin standardeihin, menetelmiin ja arkkitehtuureihin.

5.1 Kohdealueen tutkimus ja rajaaminen

Kohteena ovat terveydenhuollon tietojärjestelmät ja sovellusalueena (domain) lääkitystieto. Kohdealueen rajaamisesta tekee vaikean lääkitystiedon moninaiset käyttötavat ja tiedon hajautunut sijainti. Tästä syystä kohdealuetta on tarkennettu vielä sillä, että kes-

kitytään terveydenhuollon ammattihenkilöstön, erityisesti lääkärin näkökulmaan, jonka kautta voidaan kartoittaa merkittävä osa lääkitystiedon tarpeista. Potilasnäkökulma ja apteekki voidaan perustellusti rajata ulkopuolelle sen vuoksi, että ne eivät toistaiseksi ole suorassa kosketuksessa terveydenhuollon tietojärjestelmiin ja ne tulisi mallintaa erikseen (komponenttiajattelu). Potilaan tietoja kyllä käsitellään järjestelmissä, mutta siihen ei osallistu potilas vaan vastuussa on hoitava henkilökunta. Apteekkien rajapinnoista merkittävin lienee tuleva sähköinen resepti, mutta sille on jo olemassa määritykset. Myös sairaala-apteekki voidaan rajata omaksi komponentiksi. Apteekkien kanssa tapahtuva tiedonvaihto on kuitenkin yksi rajapinnoista, joiden tulisi näkyä tietoarkkitehtuurissa sitä myöhemmin laajennettaessa.

5.2 Lääkitystiedon prosessimalli

Prosessimallinnusvaiheessa analysoitiin terveydenhuollon hoitoprosesseja terveydenhuollon ammattilaisen, erityisesti lääkärin, sekä tietojärjestelmän näkökulmasta. Mallien fokus on tiedossa ja sen käytössä. Kuvaus keskittyy niihin hoitoprosessin osiin, joissa käsitellään lääkitystietoa, ne eivät siten kuvaa kattavasti potilaan koko hoitoprosessia. Myöskään potilaan kotona itse tai muiden antamana saatua hoitoa ei kuvata. Lääkitystiedon rakennetta ja sisältöä selvitettiin analysoimalla tiedon kulku, tiedon saanti, tiedon käyttö ja lääkitykseen liittyvän tiedon rooli hoitoprosessissa.

Liitteessä 2 on nähtävillä tuotetut prosessimallit, joissa ylimmällä tasolla prosessi jakautuu neljään vaiheeseen. Se alkaa hoidon tarpeen arvioinnista potilaan saapuessa lääkärin tai hoitajan vastaanotolle ja päättyy potilaan päästessä kotiin tai siirryttäessä jatkohoitoon, jolloin katsotaan käynnistyvän uusi hoitoprosessi. Ylätason vaiheet ovat iteratiivisia, samoja vaiheita toistetaan kunnes hoidon tavoitteet on saavutettu tai potilas siirtyy jatkohoitoon muualle.

5.3 Lääkitystiedon tietomalli

Lääkitystiedon tietomalli (Liite 3) on tehty ER-mallinnusmenetelmällä muun muassa siksi, että sitä pidettiin helpoimmin omaksuttavana myös tietojärjestelmätieteisiin perehtymättömälle. Syntyneitä tietomallia tarkennettiin asiantuntijaläpikäyntien avulla. [Nykänen ja muut, 2008]

Tietomalli jakautuu kahteen osaan: 1) Lääkitystiedon konteksti ja 2) Lääkitystietomerkintä. Lääkitystiedon konteksti ei ole osa lääkitystietoa, mutta kuvaa lääkitystietomerkinnän suhteessa muun muassa sähköiseen reseptiin ja kansallisten määrittelyiden asiakirjaperustaiseen malliin. Kuvaus on hyvin yleisluontoinen, eikä välttämättä ole yhdenmukainen kansallisten määrittelyiden kanssa, jonka asiakirjaperustaisen mallin yksityiskohtiin ei tässä yhteydessä syvennytä. Todettakoon kuitenkin tietomallin kuvauksessa ajatuksena olevan, että 0–n Lääkitystietomerkintää syntyy jossakin Palvelutapahtu-

massa, ja että Palvelukokonaisuuteen liittyy 1–n Palvelutapahtumaa. Hoitoprosessiin voi liittyä 1–n Palvelukokonaisuutta ja -tapahtumaa. Näin ollen olisi mahdollista tarkastella esimerkiksi tiettyyn palvelukokonaisuuteen tai tiettyihin palvelutapahtumiin liittyviä lääkityksiä. Muutokset lääkityksessä ajoittuvat palvelutapahtumiin liittyvien lääkitystietomerkin­töjen kirjausajankohtiin.

Lääkitystietomerkin­tä on Merkinnän osa. Lääkitystietomerkin­nällä on tekijä (Terveydenhuollon ammattihenkilö), kohde (Potilas) ja merkintään voi liittyä myös sähköinen tai muu resepti. Mallissa ei tarkoituksellisesti käytetä käsitettä lääkemääräys, sillä sen tulkitaan usein tarkoittavan reseptiä ja merkinnässä on kuitenkin kyse eri asiasta – se voi olla vaikkapa osastolla annettua lääkitystä, josta reseptiä ei tuoteta. [Nykänen ja muut, 2008]

Tietomallin ensimmäinen osa, lääkitystiedon konteksti, on tarpeellista purkaa tässä vain osittain, koska se ei ole varsinainen osa lääkitystiedon tietomallia. Muun muassa sähköiselle lääkemääräykselle on jo olemassa omat määrittelynsä [STM, 2008]. Lääkitystiedon kontekstista tietoarkkitehtuuriin on otettu mukaan Potilas, Terveydenhuollon ammattihenkilö, Merkin­tä ja Resepti (sähköinen tai muu resepti).

5.4 Lääkitystiedon harmonisointi viitemalleihin

Lääkitystiedon tietomalli on sovellus- tai kohdealueen tietomalli, jonka lähin vastine HL7-kehitysmen­etelmässä on D-MIM. D-MIM johdetaan HL7 RIM:stä seuraavasti [Lopez and Blobel, 2006; HL7 V3, 2009]:

- Vaihe 1.** Kohdealueen luokat jaotellaan RIM-ydinluokkien mukaisesti: toiminto, entiteetti tai rooli
- Vaihe 2.** Kohdealuetta verrataan nykyisiin HL7-tietomalleihin ja valitaan kohdealueeseen sopivat. Kandidaatteja tähän ovat erilaiset D-MIM, R-MIM ja CMET-mallit.
- Vaihe 3.** Valitaan ja nimetään tarvittavat HL7 RIM -luokat. Jos samasta luokasta tarvitaan enemmän kuin yksi kopio, se kloonataan ja annetaan kloonille kuvaava nimi. Sen jälkeen lisätään kloonien välille suhteet RIM-mallin suhteiden logiikkaa noudattaen. Luokka voi sisältää samat attribuutit tai osan niistä kuin alkuperäinen RIM luokka, minkä lisäksi se luonnollisesti perii vanhempansa attribuutit.
- Vaihe 4.** Valitaan attribuutteja varten tarvittavat datatyypit ja sanastot ja liitetään ne kuhunkin attribuuttiin.

HL7 RIM -luokkien attribuuttien avulla ja kloonamalla sen luokkia tarpeellinen määrä pitäisi olla mahdollista kuvata kaikki lääkitystiedon tietomallissa esiintyvät attri-

buutit. Lisäksi RIM ja siitä johdetut tarkennetut mallit elävät jatkuvasti ja niitä muutetaan ja laajennetaan tarvittaessa äänestysprosessin kautta [HL7 V3, 2009], joten mahdolliset muutostarpeet ovat ratkaistavissa. RIM:ssä on käytettävissä suuri määrä erilaisia tietotyyppejä, ja sopivia attribuutteja sisältäviä luokkia valitsemalla ja kloonamalla saa lääkitystiedon tietomallin luokat ja attribuutit avattua. Lääkitystiedon tietomallin käsitteistö ei ole yhtenevä kansallisten luokitusten ja määritysten kanssa, mikä on yksi syy siihen, että attribuuttien purkaminen HL7 RIM:n mukaiseksi olisi hyvin työläs tehtävä. Tässä yhteydessä esitän osittaisen harmonisoinnin, joka keskittyy todentamaan tietomallin muunnettavuuden HL7 RIM -mallin mukaiseksi. Harmonisoinnissa edetään siten tasolle 2 ja osittain tasolle 3.

Vaihe 1: Kohdealueen luokkien jaottelu

Jaotteluun on tässä haettu perustaa Medication D-MIM:sta (POME_DM000000, v1.4.7), joka on nähtävissä liitteessä 4. Kyseinen D-MIM on rakennettu siten, että lääkitys kuvataan roolina, johon liittyy lääkettä kuvaavia entiteettejä (vahvuus, ainesosat ym.) sekä useita toimintoja (annostusohje, hinta ym.). Toimintoihin kytkeytyy rooleja, kuten potilas ja toiminnon toteuttaja. Esimerkiksi AdministrationGuideline-toiminnolla on sekä tekijä (author) että kohde (subject), joka on potilas. Medication D-MIM on monelta osin erilainen kuin lääkitystiedon tietomalli, mutta erityisesti kuvattu periaate – lääkitys roolina – kannattaa hyödyntää, sillä lääkitykseen liittyy aina tekijä, kohde, lääke sekä määräämiseen liittyvät toiminnot. Luokkien jaottelu on esitelty taulukossa 3.

Taulukko 3: Lääkitystiedon tietomallin luokkien jaottelu HL7 RIM -perusluokkiin

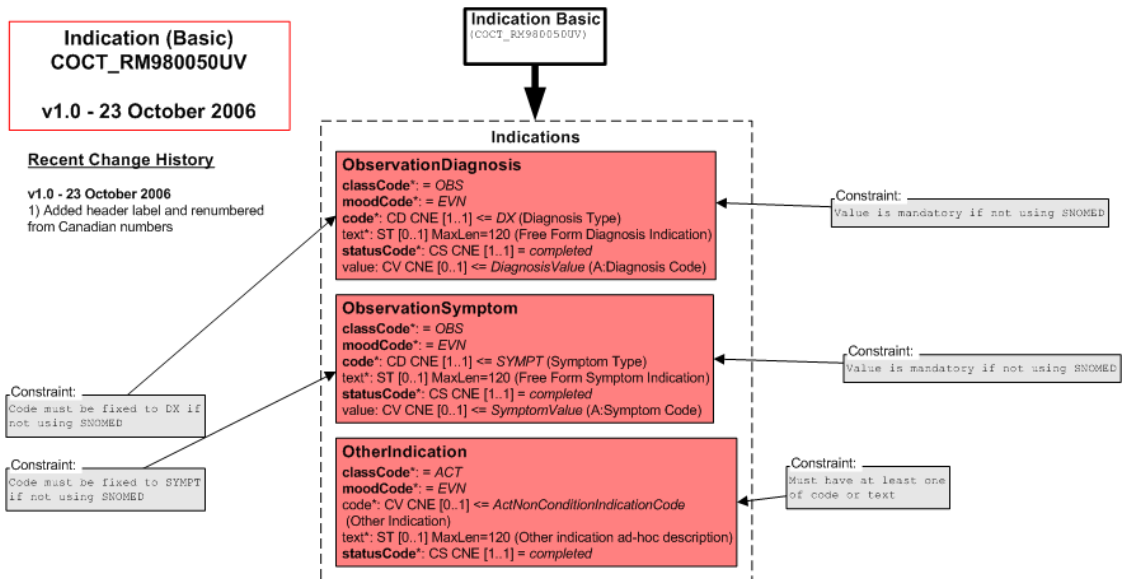
Kohde/luokka	Rooli	Toiminto	Entiteetti
Lääkitystiedon konteksti			
Potilas	*		
Terveystieteiden ammattihenkilö	*		
Resepti (sähköinen tai muu)		*	
Merkintä		*	
Lääkitystietomerkintä			
Lääkitysyhteen veto		*	
Lääkitys	*		
Vaikutus		*	
Lääkityksen muutos		*	

Kohde/luokka	Rooli	Toiminto	Entiteetti
Lopettaminen		*	
Sivuvaikutukset		*	
Uusinta		*	
Toteutunut annostelu		*	
Lääkevalmiste	*		
Muut ainesosat			*
Kela-korvattavuus			*
Laji			*
Pakkaustiedot			*
Lääkevalmisteen muutos		*	
Annostus		*	

Vaihe 2: Kohdealueeseen soveltuvat HL7-määrittelyt

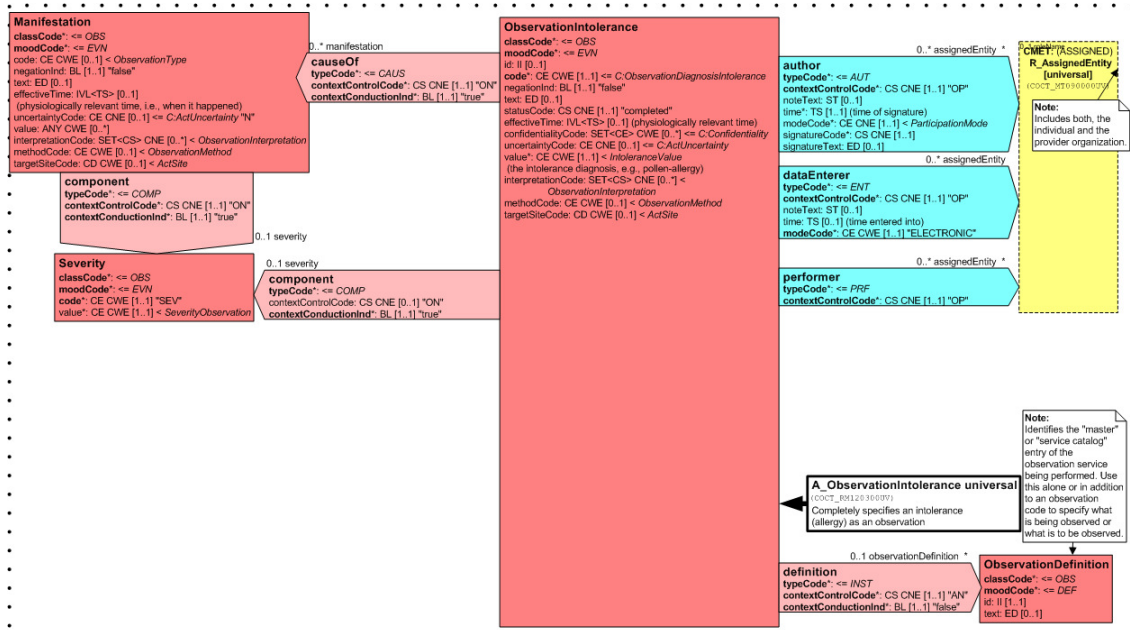
Sovellusalueeseen liittyviä HL7-viitemalleja on useita. Sellaista tarkennettua viitemallia ei ole löytynyt, johon lääkitystiedon tietomallin voisi sovittaa ilman suuria muutoksia. Lääkitykselle on olemassa Medication D-MIM (Liite 4) ja lisäksi löytyy useita tietomalliin osittain sopivia CMET-määrittelyjä. Parhaan lähtökohdan lääkitystiedon tietoarkkitehtuurille antaa Medication D-MIM (POME_DM000000, v1.4.7). Harmonisointi on tehty käyttäen HL7 V3 ballot May 2009:ää [HL7, 2009c] ja HL7 RIM -versiota 2.26 (Liite 1).

Monet tietomallin attribuuteista täytyy ilmaista omana luokkana, koska kussakin HL7 RIM -luokassa on käytettävissä vain tietyt attribuutit. Luokan attribuuteista voi käyttää vain osaa, mutta niitä ei voi lisätä. Siksi luokan laajentaminen on tehtävä kloonamalla sopivaa luokkaa tarpeellinen määrä. Esimerkiksi Terveystieteiden ammattihenkilöä ei voi ilmaista yksittäisellä HL7 RIM -luokalla, vaan muun muassa sen attribuutti Organisaatio ilmaistaan aliluokalla Organization. Käytännössä kaikissa tietomallin luokissa on attribuutteja, jotka tulisi esittää omana luokkana. Asiaa mutkistaa vielä sekin, että esimerkiksi Lääkitys-luokan attribuutin Indikaatio/Diagnoosi/Hoidon tavoite ilmaisemiseen mahdollisesti tarvittaisiin paitsi oma luokka myös aliluokkia ja viittauksia muissa malleissa määriteltyihin potilaan tietoihin. HL7 RIM -viitemalleista löytyy ainakin osittain soveltuvia CMET-määrittelyjä (mm. CMET A_Indication [basic] ja CMET R_Patient [universal]), mutta täydellistä vastaavuutta suomalaisiin luokituksiin ja käsitteistöön ei välttämättä ole. Kuvassa 18 on CMET v1.0 indikaatiolle.



Kuva 18: CMET Indication (Basic) – Indikaation CMET [HL7, 2009c]

Kuvassa 19 on CMET A_ObservationIntolerance universal (COCT_RM120300UV), jolla voidaan esittää jonkin lääkityksen sivuvaikutukset. Taulukossa 4 on haettu vastaavuuksia lääkitystiedon tietomallin Sivuvaikutukset-luokan attribuuttien ja A_ObservationIntolerance universal CMET:in välillä. Tuloksena on, että kaikille attribuuteille löytyy soveltuva attribuutti tai luokka CMET:stä. Samalla periaatteella voisi hakea vastaavuuksia myös muista luokista ja ottaa siten käyttöön tarjolla olevat valmiit palaset – tai havaita niissä mahdolliset muutostarpeet.



Kuva 19: CMET A_ObservationIntolerance universal – Sivuvaikutukset [HL7, 2009c]

Taulukko 4: Lääkitystiedon tietomallin sivuvaikutukset-luokka ja CMET A_ObservationIntolerance [HL7, 2009c]

Lääkitystiedon tietomalli	A_ObservationIntolerance universal (COCT_RM120300UV)		
<table border="1" style="width: 100%;"> <tr> <td style="text-align: center; background-color: #cccccc;">Sivuvaikutukset</td> </tr> <tr> <td>Koettu Sivuvaikutus Sivuvaikutuksen varmuusaste Sivuvaikutuksen todennut lääkäri Sivuvaikutuksen todenneen lääkärin organisaatio Koska todettu</td> </tr> </table>	Sivuvaikutukset	Koettu Sivuvaikutus Sivuvaikutuksen varmuusaste Sivuvaikutuksen todennut lääkäri Sivuvaikutuksen todenneen lääkärin organisaatio Koska todettu	<ul style="list-style-type: none"> • Koettu sivuvaikutus: Manifestation • Sivuvaikutuksen varmuusaste: ObservationIntolerance -> uncertaintyCode: CE CNE [0..1] <= C:ActUncertainty • Sivuvaikutuksen todennut lääkäri, sivuvaikutuksen todenneen lääkärin organisaatio: CMET: (ASSIGNED) R_AssignedEntity (Universal) • Koska todettu: author -> time: TS [1..1] (time of signature)
Sivuvaikutukset			
Koettu Sivuvaikutus Sivuvaikutuksen varmuusaste Sivuvaikutuksen todennut lääkäri Sivuvaikutuksen todenneen lääkärin organisaatio Koska todettu			

Harmonisointiin voisi käyttää myös CDA R2:n määrittelyjä ja Suomeen lokalisoitua lääkityslistan CDA R2 -määrittelyä. Tavoitteena ei kuitenkaan ole sovittaa tietomallia CDA R2 -rakenteeseen, joten lääkityslistan CDA R2 jää taustamateriaaliksi. Sitä on

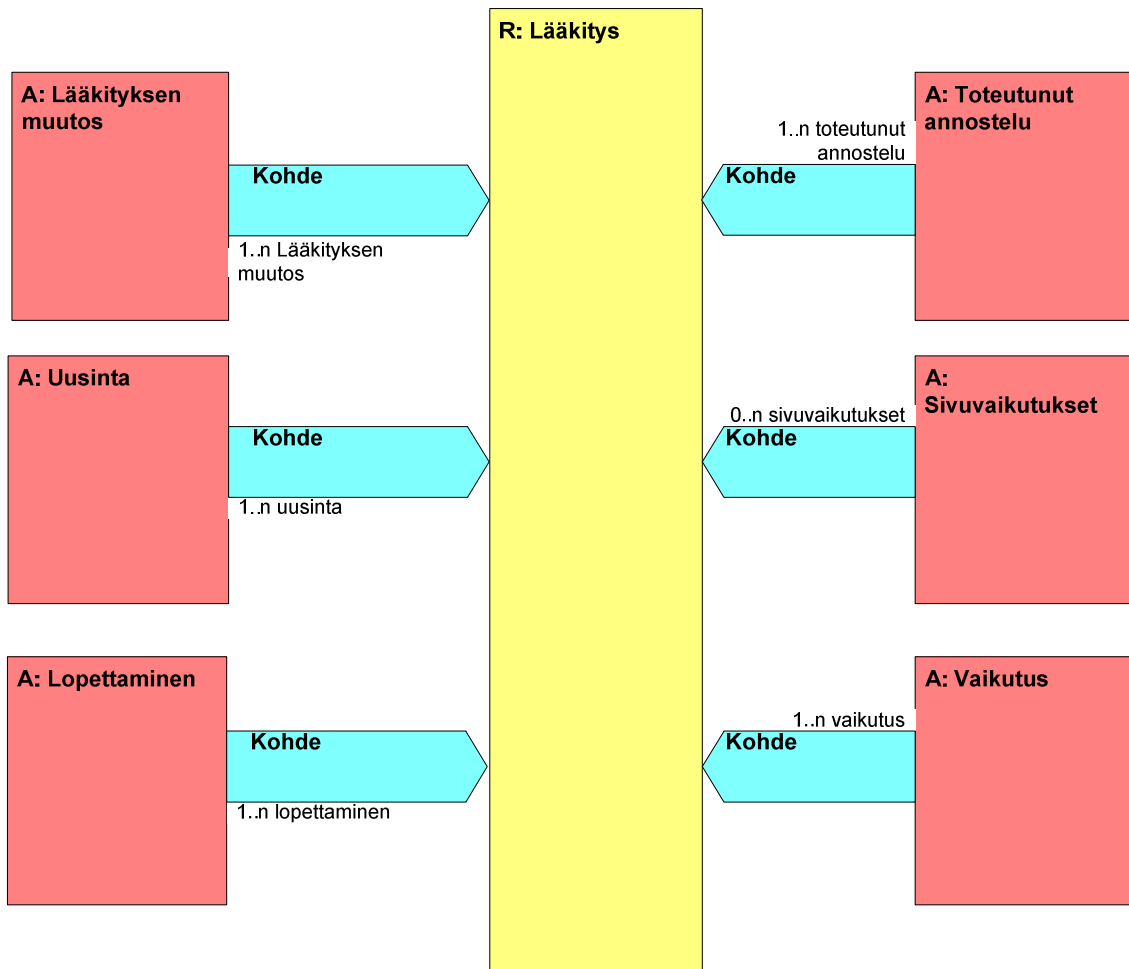
aiemmin käytetty yhtenä lääkitystiedon tietomallin referenssinä, jonka myötä tietomallin tietosisältö on pitkälti yhteensopiva ja todennäköisesti ilmaistavissa CDA-määrittelyn mukaisena kohtuullisen ongelmattomasti.

5.5 Lääkitystiedon tietoarkkitehtuuri

Lääkitystiedon tietomalli on mahdollista muokata HL7 RIM perusluokkien mukaiseksi varsin sujuvasti.

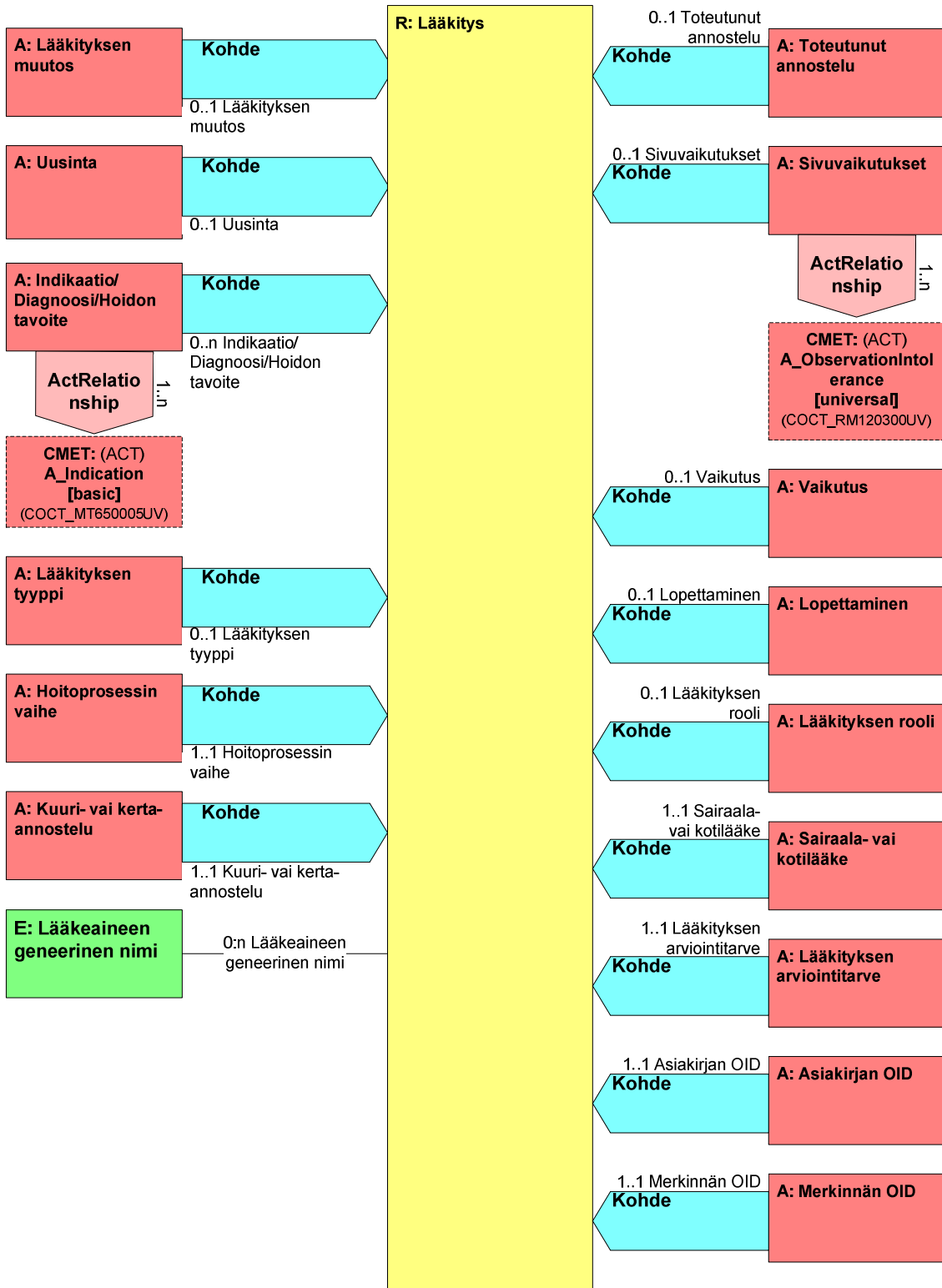
Medication D-MIM:n tapaan Lääkitys esitetään roolina (Kuva 20). Toinen vaihtoehto olisi käyttää toimintoluokkaa. Rooli on kuitenkin luontevampi valinta, koska lääkityksen voi ajatella eräänlaisena abstraktiona, johon kytkeytyy konkreettisia toimintoja tai tapahtumia, kuten lääkkeen määrääminen, lopettaminen ja uusiminen. Tapahtumiin liittyy entiteettejä, kuten lääke, kohteena oleva potilas ja tapahtuman tuottajana toimiva lääkäri. Lääkityksen esittäminen roolina pitää syntyvän hierarkian myös matalampana, koska se pystyy sitomaan suoraan ympärilleen lähes kaikki alkuperäisenkin tietomallin luokat ilman Act Relationship tai Role Link -luokkia. Kuvassa 20 on aloitettu HL7 RIM:n mukaisen tietoarkkitehtuurin rakentaminen kuvaamalla Lääkitys-luokka ja sen naapuriluokat lukuun ottamatta Lääkitysyhteenvedoa ja Lääkevalmistetta.

Luokkakaavioiden (Kuvat 20–22) esitystapa mukailee HL7 v3 Guiden ohjeita ja UML:ää, mutta ne eivät ole varsinaisia D-MIM-kaavioita eivätkä myöskään sellaisenaan toteuttamiskelpoisia. Värikoodauksen lisäksi luokkien nimen eteen on lisätty kirjain A, R tai E. Luokkien välillä nähtävät nuolet ovat Osallisuus (Participation) (vaakasuuntaisia), Act Relationship ja Role Link -luokkia.



Kuva 20: Lääkitys ja siihen liittyviä toimintoja

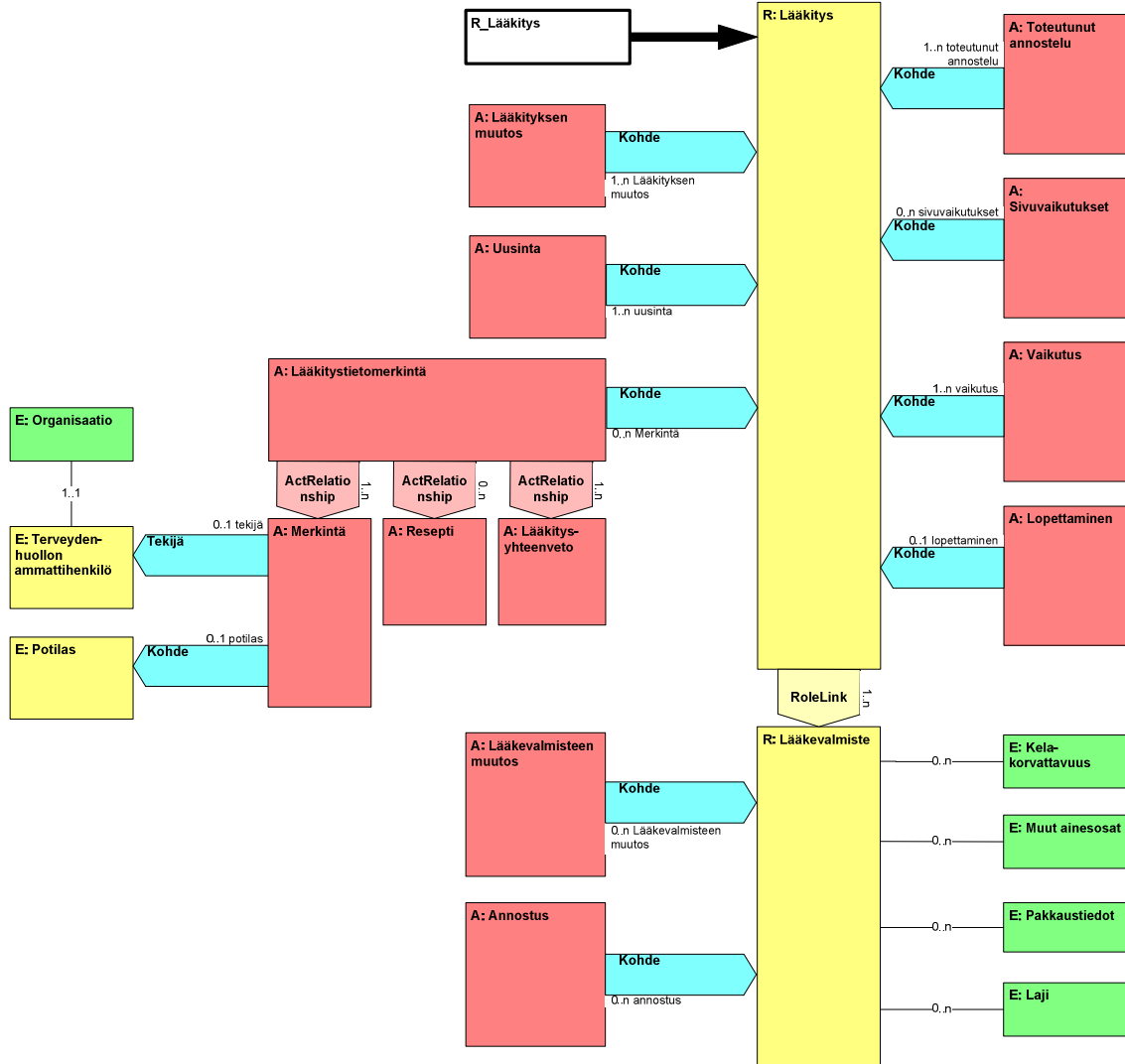
Kuvassa 21 on otettu esimerkiksi Lääkitys-luokka siitä, millaisia uusia luokkia sen attribuutit voisivat synnyttää. Niitä ei tässä ole kuitenkaan määritetty ydinluokkaa tarkemmin, sillä se vaatisi täsmällistä tietämystä kunkin attribuutin esitystavasta ja tietosisällöstä. Osa attribuuteista pilkkoutuu kuvassa esitettyä pienempiin osiin, joidenkin esittäminen vaatii useita luokkia. Toisaalta sama luokka saattaa pystyä esittämään kerralla useita attribuutteja. Kuvassa on kaksi viittausta CMET-malleihin. Kyseiset mallit on valittu mukaan siksi, että niillä näyttäisi pystyvän kuvaamaan melko hyvin kyseisen attribuutin sisällön. Ne ovat kuitenkin tarkoitettu esimerkeiksi, eivätkä todennäköisesti olisi sellaisenaan soveltuvia. Seuraava vaihe olisi attribuuttien vertaaminen kansallisiin määrittelyihin ja luokituksiin ja tarkempien HL7 RIM -luokkien valitseminen.



Kuva 21: Lääkitys-luokka ja sen attribuutit

Kuvassa 22 on mukana kaikki tietomallin luokat lukuun ottamatta lääkitystiedon kontekstista pois rajattuja luokkia. Edellä kuvassa 21 esitetyt Lääkitys-luokan attribuutit on jätetty pois. Joitakin muutoksia alkuperäisen tietomallin rakenteeseen on tehty, mutta kaikki luokat ja tietosisältö säilyttäen. Palvelutapahtuma ja Palvelukokonaisuus kytkey-

tyisivät kuvatussa mallissa Act Relationship -luokalla luokkaan A: Merkintä. Valkoinen laatikko R_Lääkitys on tarkennetun Lääkitys R-MIM -luokan entry point.



Kuva 22: Lääkitystiedon tietomalli harmonisoituna HL7 RIM:n perusluokkiin

6 Yhteenveto ja johtopäätökset

Tutkimuksessa tutustuttiin aluksi kansalliseen ja kansainväliseen terveydenhuollon tietojärjestelmien nykytilaan ja suunnitelmiin erityisesti lääkitystiedon näkökulmasta. Ongelmat lääkitystiedon saatavuudessa ja luotettavuudessa on tunnistettu kaikissa tutkituista maista, mutta ratkaisuja ei ole toistaiseksi haettu kovin laaja-alaisesti vaan lähinnä rakentuen sähköisen reseptin ympärille. Myös Suomen KanTa-arkkitehtuurissa ja erilaisissa kansallisissa määrittelyissä paino on lääkemääräyksiin perustuvassa lääkitystietojen käsittelyssä. Vallitseva lainsäädäntö ja erilaiset määritykset antavat perustan organisaatioiden väliselle tiedonvälitykselle, mutta niissä on myös rajoituksensa.

Läkitystietoa tallennetaan ja käytetään monin eri tavoin. Sitä voi myös syntyä useassa eri paikassa, ja terveydenhuollon ammattilaiselle olisi tärkeää saada käyttöönsä eri organisaatioissa syntynyt tieto. Eri yhteyksissä lääkitystietoa tulisi voida tarkastella monista eri näkökulmista ja eri abstraktiotasoilla. Joskus tietoa tarvitaan diagnoosiin tai indikaatioon perustuen, joskus tarvitsee nähdä tarkasti, millä kauppanimillä lääkettä on määrätty. Lääkitystä halutaan tarkastella myös vaikuttavan aineen perusteella. Hyvä lääkitystiedon hallintamalli on sellainen, joka tyydyttää mahdollisimman monipuolisesti eri osapuolten tiedonkäsittelytarpeet mukaan lukien kansalaisen ja apteekin. Tällaisen hallintamallin perusedellytys on, että lääkitystieto tallennetaan yhdenmukaisesti ja rakenteellisesti. Tietojärjestelmien kehittyessäkin kokonaislääkityksen muodostamisessa potilaalla tulee olemaan tärkeä asema.

Terveydenhuollon tietojärjestelmiin liittyvän tietoarkkitehtuurin suunnitteluun on olemassa monia hyviä vaihtoehtoja, joista muutamia käsiteltiin tässä tutkimuksessa. Eri suunnittelumenetelmien sisältämiä vaiheita tutkittiin ja koostettiin niiden perusteella nelivaiheinen tietoarkkitehtuurin suunnitteluprosessi.

Tutkimuksessa sovellettiin tietoarkkitehtuurin suunnitteluprosessissa HL7 RIM:iä ja HDF-menetelmää ja johdettiin lääkitystiedon tietomallista HL7 RIM -perusluokkiin harmonisoitu tietoarkkitehtuuri. Tutkimuksen viitemalli ja suunnittelumenetelmä valikoituivat lopulta helposti. Ensinnäkin kansallisessa kontekstissa on luonnollista pyrkiä huomioimaan, mitä ratkaisuja terveydenhuollon tietojärjestelmissä tällä hetkellä käytetään ja mikä on tulevaisuuden trendi. Suomessa sellainen on keskitetyn kansallisen arkiston tuleva käyttöönotto. Toisekseen HL7 RIM on kypsä viitemalli, jonka luokat, niiden attribuutit ja tietotyypit sekä hyväksymisprosessin läpikäyneet tai siinä olevat D-MIM-, R-MIM- ja CMET-määritykset tukevat harmonisointiprosessia hyvin. Valittu menetelmä osoittautui toimivaksi kohdealueen tietoarkkitehtuurin tuottamisessa ja soveltuu käytettäväksi vastaavanlaisiin tapauksiin, joissa kohdealueena on jokin terveydenhuollon osa-alue ja joissa HL7 RIM:n käyttö on perusteltua.

Tietoarkkitehtuurin perusrakenteesta syntyi HL7 RIM -perusluokkia käyttämällä selkeä ja johdonmukainen. Lopputuloksena on harmoninen kokonaisuus, joka tukee lääkitystiedon erilaisia käyttötilanteita ja mahdollistaa tietojen tarkastelun monista eri näkökulmista.

Viiteluettelo

- [Aarnio, 2007a] Eeva Aarnio, Lääkitystiedon käsite ja sidonnaisuudet. Lääkitystiedon hallinta -hanke, osatehtävä 2, raportti, Turun kauppakorkeakoulu, 2007.
- [Aarnio, 2007b] Eeva Aarnio, Lääkitystietojen jakaminen kansallisella ja alueellisella tasolla. Lääkitystiedon hallinta -hanke, osatehtävä 2, raportti, Turun kauppakorkeakoulu, 2007.
- [Aarnio, 2008a] Eeva Aarnio, Terveystieteiden ammattilaiset ja lääkitystiedon tarpeet – Lääkitystiedon lähteet perusterveydenhuollossa. Teoksessa *Lääkitys kohdalleen – Tietojen hallinta avain oikeaan lääkitykseen*, Reima Suomi, Reetta Raitoharju (toim.), Turku Centre for Computer Science, TUCS National Publication no 16, 2008, 25–32.
- [Aarnio, 2008b] Eeva Aarnio, Lääkitystiedon hallintamallit – Terveystieteiden ammattilaisen näkökulma. Turun kauppakorkeakoulu, Turku, 2008.
- [Blobel, 2007] Inventory of Relevant Standards for EHR Systems. Version 0.8. Q-REC, IST-27370-SSA, European Commission – DG Information Society. ProRec Germany, eHealth Competence Center, University of Regensburg Medical Center, 10. January 2007. Available as: http://www.eurorec.org/services/standards/Q-REC-Eurorec_Overview%20on%20EHR%20Standards%2001.pdf, checked: 10.4.2009.
- [Brancheau and Wetherbe, 1986] James C. Brancheau and James C. Wetherbe, Information architectures: methods and practice. *Information Processing and Management* **22**, 6 (Dec. 1986), 453–463.
- [CEN EN 12967-2:2007] Health Informatics – Service architecture – Part 2: Information viewpoint, ISO/TC 251, CEN EN 12967-2:2007.
- [Chen, 1976] Peter Chen, The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems* **1**, 1 (Mar. 1976), 9–36.
- [Cisneros et al., 1997] Laura Cisneros, David Hunt and Donald McCollam, Information architecture – bring the university’s information inventory under control. Proceedings of CAUSE’97 conference, available as: <http://net.educause.edu/ir/library/html/cnc9759/cnc9759.html>, checked: 26.7.2009.

- [Denhardt and Denhardt, 2000] Robert Denhardt, Janet Denhardt, The New Public Service: Serving Rather than Steering. *Public Administration Review* **60**, 6, 549–559.
- [eHealth ERA, 2007] eHealth – priorities and strategies in European countries. eHealth ERA report- March 2007. European Commission, Information Society and Media, European Communities, 2007.
- [EuroRec, 2006] EuroRec, An inventory of certification criteria for EHR systems, Available as: <http://www.eurorec.org>, checked: 15.12.2006.
- [Gilchrist and Mahon, 2004] Alan Gilchrist (ed.) and Barry Mahon (ed.), *Information architecture. Designing information environments for purpose*. Facet Publishing, London, 2004.
- [HDF, 2005] HL7 Development Framework Chapter 2 – Requirements documentation, Modeling and analysis, version 2.0.0. 11.5.2005. Available as: http://hdf.wikispaces.com/file/view/HDFChapter2-RequirementsDocumentation_v4.doc. Checked: 10.4.2009.
- [HE, 2006] Hallituksen esitys n:o 250 Eduskunnalle laiksi sähköisestä lääkemääräyksestä sekä laiksi lääkelain 57 ja 57 a §:n muuttamisesta. HE n:o **250/2006** vp. Saatavissa: <http://www.finlex.fi/fi/esitykset/he/2006/20060250>, tarkistettu 26.7.2009.
- [Hevner et al., 2004] A. R. Hevner, S. T. March, J. Park, S. Ram, Design Science in information systems research and practice: The reflective practitioner as a researcher, In: Kumar and DeGross (Eds.) Proc. of the 18th ICIS conference, ACM, New York, 121–131.
- [HIS-DF] Health Information Systems Development Framework website. Available as: <http://www.gist.det.uvigo.es/~dmlopez/HIS-DF/>. Checked: 10.4.2009.
- [HL7, 2009a] Health Level Seven. Www-sivu, <http://www.hl7.org>, Checked: 26.7.2009.
- [HL7, 2009b] Health Level Seven, HL7 Reference Information Model 2.26, Available as: <http://www.hl7.org/library/data-model/RIM/C30226/rim0226c.zip>, Checked: 26.7.2009

- [HL7, 2009c] Health Level Seven Version 3.0 May 2009 Ballot.
- [HL7 V3, 2009] Health Level Seven, *HL7 V3 Guide*, HL7 V3 ballot May 2009.
- [Hoffer et al., 1999] Hoffer Jeffrey A., George Joey F., Valacich Joseph S. *Modern Systems Analysis & Design*. Second edition. Addison-Wesley, 1999.
- [Hyppönen ja muut, 2005] Hannele Hyppönen (toim.), *Sähköisen reseptin pilotoinnin arviointi vaihe I – Loppuraportti*, Osaavien keskusten verkoston julkaisuja 1/2005, Helsinki.
- [Jong, 2007] Tom de Jong, The Electronic Medication Record in The Netherlands. A co-production by NICTIZ – HL7 the Netherlands – Healthcare Stakeholders – IT industry. Presentation at Tekes Interoperability Seminar, Helsinki, Feb 14th 2007. Available as: http://akseli.tekes.fi/opencms/opencms/OhjelmaPortaali/ohjelmat/FinnWell/fi/Dokumenttiarkisto/Viestinta_ja_aktivointi/Seminaarit/INTEROPERABILITY/Electronic_Medication_Record_in_the_Netherlands.ppt, checked: 26.7.2009.
- [Järvinen & Järvinen, 2004] Pertti Järvinen ja Annikki Järvinen, Tutkimustyön metodeista. Opinpajan kirja, 2004.
- [Karimaa, 2001] Karimaa Erkki, *Osaaminen on malleissa*. Suomen Kuntaliitto, 2001.
- [Laki 61/2007] Laki sähköisestä lääkemääräyksestä, 2.2.2007/61. Saatavissa: <http://www.finlex.fi/fi/laki/ajantasa/2007/20070061>, tarkistettu: 26.7.2009.
- [Laki 159/2007] Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä, 9.2.2007/159. Saatavissa: <http://www.finlex.fi/fi/laki/ajantasa/2007/20070159>, tarkistettu: 26.7.2009.
- [Larson et al., 2003] Mats Larson, Gustav Malmqvist, *Sjunet – The National IT Infrastructure for Healthcare in Sweden*. In: An International Conference of the Arctic Council, Akureyri, Iceland, 2003. Available as: <http://vefir.unak.is/ICTConference/Larson.pdf>, Checked: 26.7.2009.
- [Lopez and Blobel, 2006] Diego M. Lopez, Bernd Blobel, Formal Design of Electronic Public Health Records. In: L. Bos, L. Roa, K. Yogesana, B. O’Connell, A. Marsh, B. Blobel (Eds.), *Medical and Care Computation 3*, Series Studies in Health Technology and Informatics, **121**, IOS Press, Amsterdam, 2006, 337–348.

- [Lopez and Blobel, 2009] Diego M. Lopez, Bernd G.M.E. Blobel, A development framework for semantically interoperable health information systems. *International Journal of Medical Informatics* **78**, 2009, 83-103.
- [March and Smith, 1995] S. T. March, G. F. Smith, Design and natural science research on information technology, *Decision Support Systems* **15**, 251–266.
- [Mäki, 2008] Ilkka Mäki, *Tapaustutkimus kansallisen sähköisen arkiston ja viitetietokantaan perustuvan aluetietojärjestelmän yhteensovittamisesta*. Pro gradu - tutkielma, Tampereen yliopisto, Tietojenkäsittelyoppi, 2008. Saatavana: http://www.cs.uta.fi/research/theses/masters/Maki_Ilkka.pdf, tarkistettu: 26.7.2009.
- [Mäkinen ja muut, 2008] Joonas Mäkinen, Pirkko Nykänen, Antti Toivanen, *Osaraportti 2: Lääkitystiedon tietomalli*. Lääkitystiedon hallinta -hankkeen sisäinen raportti, Tampereen yliopisto, 2008.
- [Mäkinen ja Nykänen, 2008] Joonas Mäkinen, Pirkko Nykänen, *Osaraportti 1: Lääkitystiedon prosessimallinnus*. Lääkitystiedon hallinta -hankkeen sisäinen raportti, Tampereen yliopisto, 2008.
- [Mäkinen and Nykänen, 2009] Joonas Mäkinen, Pirkko Nykänen, *Process Models of Medication Information*. In: Proceedings of the 42th Hawaii International Conference on System Sciences, January 2009, 7 pages.
- [NICTIZ, 2002] National Institute for Healthcare in Netherlands, *Design of the architecture basic infrastructure for healthcare, version 2.0*. NICTIZ, 2002. Available as:
https://www.nictiz.nl/uploaded/FILES/Corporate%20communicatie%20English/Design%20of%20the%20architecture%20basis%20infrastructure%20for%20health%20care_%20December%202002.doc.pdf, checked: 26.7.2009.
- [NICTIZ, 2005] National Institute for Healthcare in Netherlands, The national e-medication record in practice - Case study 2: information exchange between the GP, the hospital and the public pharmacist. NICTIZ, 2005. Available as:
<https://www.nictiz.nl/uploaded/FILES/Corporate%20communicatie%20English/National%20e-medicationrecord.pdf>, checked: 26.7.2009.

- [Nykänen ja muut, 2008] Pirkko Nykänen, Antti Toivanen, Joonas Mäkinen, Pirkko Kortekangas, Lääkitystiedon tietomalli. Teoksessa *Lääkitys kohdalleen – Tietojen hallinta avain oikeaan lääkitykseen*, Reima Suomi, Reetta Raitoharju (toim.), Turku Centre for Computer Science, TUCS National Publication no 16, 2008, 33–59.
- [Nykänen ja Mäkinen, 2007] Pirkko Nykänen, Joonas Mäkinen, *Lääkitystiedon integrointi potilastietojärjestelmissä – Case dementiapotilaan palveluketju*. Lääkitystiedon hallinta -hankkeen raportteja 1, Turun kauppakorkeakoulu, 2007.
- [OMG, 2009] Object Management Group, Introduction to OMG's Unified Modeling Language (UML). Www-sivu, saatavissa http://www.omg.org/gettingstarted/what_is_uml.htm, tarkistettu: 26.7.2009.
- [O'Rourke & al., 2003] Carol O'Rourke, Neal Fishman and Warren Selkow, *Enterprise Architecture Using the Zachman Framework*. Thomson Course Technology, Canada, 2003.
- [Pereira and Sousa, 2003] Getting into the misalignment between Business and Information Systems. In: 10th European Conference on Information Technology Evaluation. *ECITE Press*. Available as: <http://www.ea-kye.com/upl/%7B00bec721-9634-4fe7-bf13-9c43bb89aa7c%7D.pdf>, checked: 26.7.2009.
- [Plaisant et al., 1998] Catherine Plaisant, Ben Shneiderman and Rich Mushlin, An information architecture to support the visualization of personal histories. *Information Processing & Management* **34**, 5 (1998), 581–597.
- [RM-ODP, 2009] RM-ODP wiki, available as: <http://www.rm-odp.net/>, checked 11.4.2009.
- [SerAPI, 2006] HL7 Development Framework – taustaselvitys. Versio 1.5. SerAPI-projekti, Kuopion Yliopisto, 28.3.2006. Saatavissa: http://www.serapi.fi/menetelmat/hdf_v15.pdf, tarkistettu 26.7.2009.
- [Schekkerman, 2004] Jaap Schekkerman, *How to survive in the jungle of enterprise architecture frameworks*. Trafford, 2004.
- [Shakir, 2003] Abdul-Malik Shakir, HL7 Development Framework Tutorial. Power-Point presentation. April 2003. Available as:

<http://demo.hl7.org/participate/WorkGroups/mnm/DocumentsPresentations/Presentations/HL7%20Development%20Framework%20Tutorial%20APR-2003.ppt>,
checked: 10.4.2009.

[Siltanen, 2004] Juha Siltanen, *Tietoarkkitehtuuriin perustuva sovellusintegraatiomethodi – tapaus Tieliikelaitos*. Pro gradu -tutkielma, Tampereen yliopisto, Tietojenkäsittelyoppi, 2004. Saatavana:

http://www.cs.uta.fi/research/theses/masters/Siltanen_Juha.pdf, tarkistettu:
26.7.2009.

[Sjöborg et al., 2007] Bengt Sjöborg, Tobias Bäckström, Lars-Bertil Arvidsson, Eva Andersén-Karlsson, L. Bengt Blomberg, Birgit Eiermann, Marie Eliasson, Kjell Henriksson, Lennart Jacobsson, Ulf Jacobsson, Margaretha Julander, Per-Olof Kaiser, Carina Landberg, Jonas Larsson, Björn Molin, Lars L. Gustafsson, Design and implementation of a point-of-care computerized system for drug therapy in Stockholm metropolitan health region – Bridging the gap between knowledge and practice. *International Journal of Medical Informatics* **76**, 2007, 497–506.

[STM, 2006] Sosiaali- ja terveysministeriö, *Terveydenhuollon valtakunnallisen tietojärjestelmäarkkitehtuurin periaatteet. Alueellisista ratkaisista kansalliseen kokonaisuuteen*. Sosiaali- ja terveysministeriön selvityksiä 2006:8, Helsinki.

[STM, 2007a] Sosiaali- ja terveysministeriö, Terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin määrittelyprojekti, KANTA - kokonaisarkkitehtuuri, Arkkitehtuurimäärittely, versio 1.0. 28.2.2006. Saatavissa:
http://www.kanta.fi/earkisto/kokonaisarkkitehtuuri_arkkitehtuurimaarittely.pdf,
tarkistettu: 26.7.2009.

[STM, 2007b] Opas - ydintietojen, otsikoiden ja näkymien toteuttaminen sähköisessä potilaskertomuksessa, 2007, www.stm.fi

[STM, 2007c] Sähköisen potilaskertomuksen ydintietomäärittelyt, potilaskertomuksen ydintiedot. Versio 3.3. Excel-taulukko, 2007.

[STM, 2008] Sähköinen lääkemääräys vaatimusmäärittely. Versio 2.1, 8.9.2008. Saatavissa: <http://www.kanta.fi/sub.html>, tarkistettu: 26.7.2009.

[Sundhed.dk, 2008] The Danish eHealth experience: One Portal for Citizens and Professionals. Available as:

http://www.sundhed.dk/Images/alle/redaktion/english/The_Danish_eHealth_experience.pdf, checked 20.11.2008.

[Suomi ja muut, 2006] Reima Suomi, Lauri Salmivalli, Marjaana Koironen, Lääkitystiedon hallinta, Hankesuunnitelma Tekesin FinnWell -tutkimusohjelman toiselle hakukierrokselle. 31.1.2006.

[Suomi ja Raitoharju, 2008] Reima Suomi, Reetta Raitoharju (toim.), *Lääkitys kohdalleen – Tietojen hallinta avain oikeaan lääkitykseen*. Turku Centre for Computer Science, TUCS National Publication no 16, 2008.

[Sušelj, 2006] Marjan Sušelj, Recording of Medication History – Field experiences from pilot and national implementation in Slovenia. Presentation in The World of Health IT 2006 Conference & Exhibition, 2006.

[Tyndale-Biscoe, 2002] Sandy Tyndale-Biscoe, RM-ODP Enterprise Language ISO/ITU-T 15414/X.911. Available as: http://www.itu.int/itudoc/itu-t/com17/tutorial/81999_pp7.ppt. Checked: 11.4.2009.

[Vasconcelos et al., 2003] André Vasconcelos, Pedro Sousa, José Tribolet, Information system architectures. In: *Business Excellence '03, First International Conference on Performance Measures, Benchmarking and Best Practices in New Economy*, June 2003, Guimarães, Portugal. Available as: <http://www.inesc-id.pt/ficheiros/publicacoes/1245.pdf>, checked: 26.7.2009.

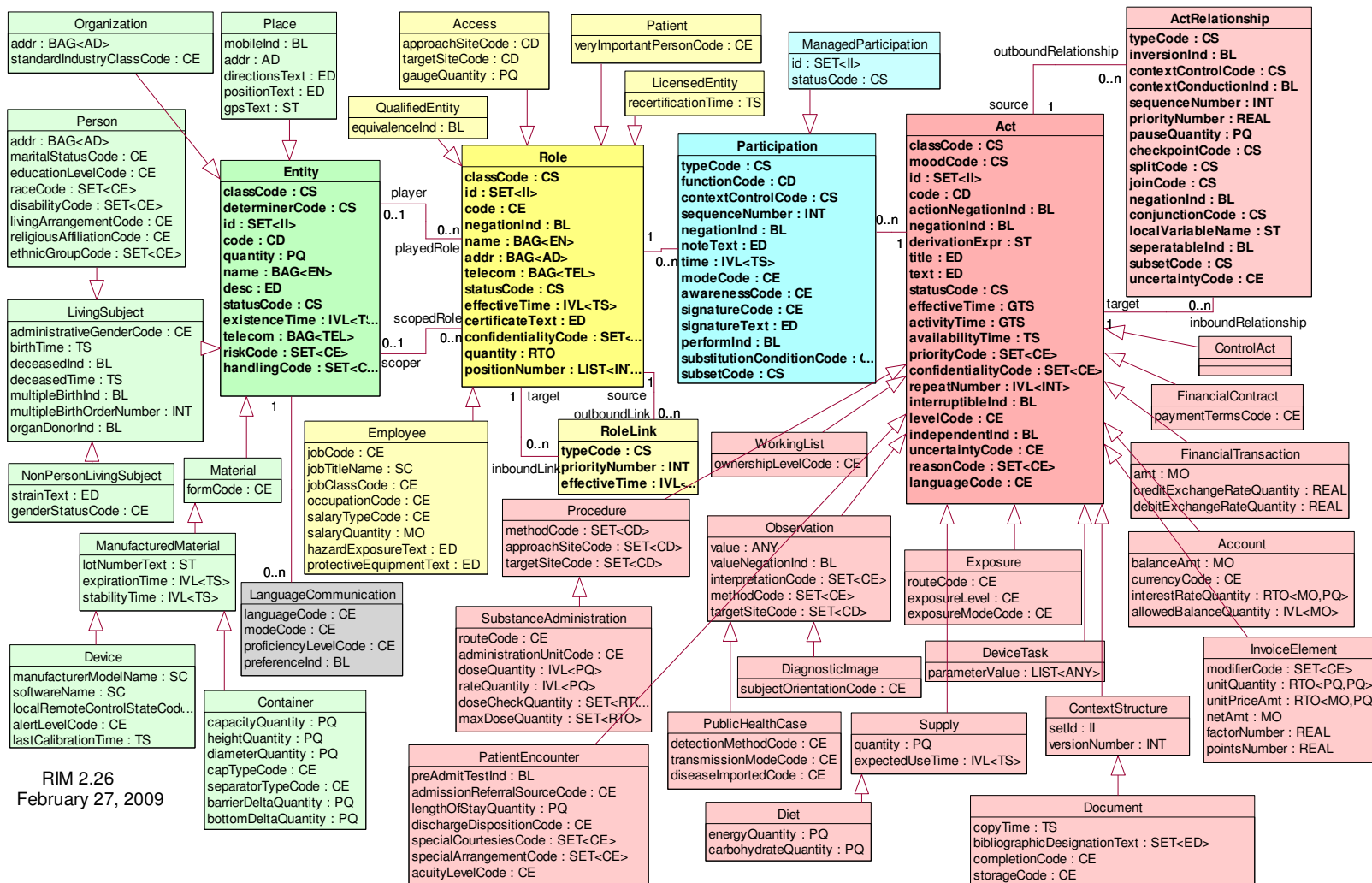
[Vasconcelos et al., 2004] André Vasconcelos, Carla Marques Pereira, Pedro Sousa and José Tribolet, Open issues on information system architecture research domain: the vision. *Proceedings of the 6th International Conference on Enterprise Information Systems*, 273–282. Available as: <http://www.inesc-id.pt/ficheiros/publicacoes/1901.pdf>, checked 26.7.2009.

[Winblad ja muut, 2008] Ilkka Winblad, Jarmo Reponen, Päivi Hämäläinen, Maarit Kangas, *Informaatio- ja kommunikaatioteknologian käyttö Suomen terveydenhuollossa vuonna 2007 – Tilanne ja kehityksen suunta*. Stakesin raportteja 37/2008, Stakes, Helsinki, 2008.

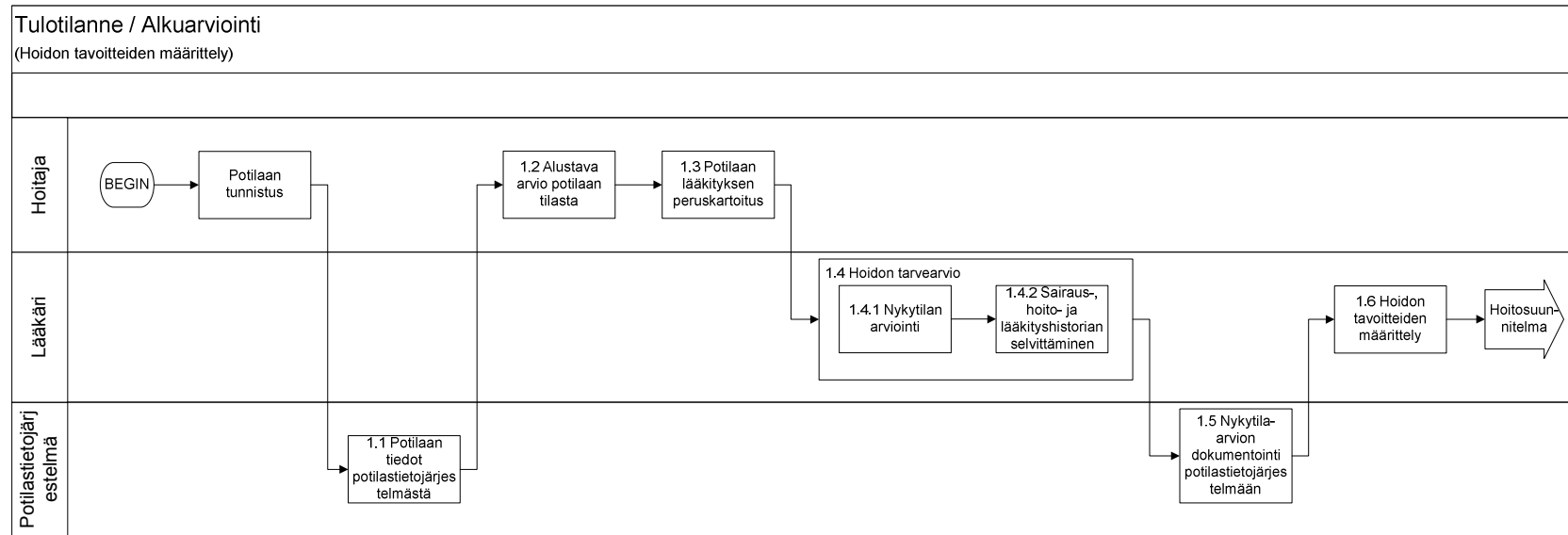
[Zachman, 1987] John A. Zachman, A framework for information systems architecture. *IBM Systems Journal* **26**, 3 (1987), 276–292.

[Zachman, 2003] John A. Zachman, *The Zachman Framework: Primer for Enterprise Engineering and Manufacturing*. Zachman International, 2003.

Liite 1: HL7 RIM versio 2.26 [HL7, 2009b]

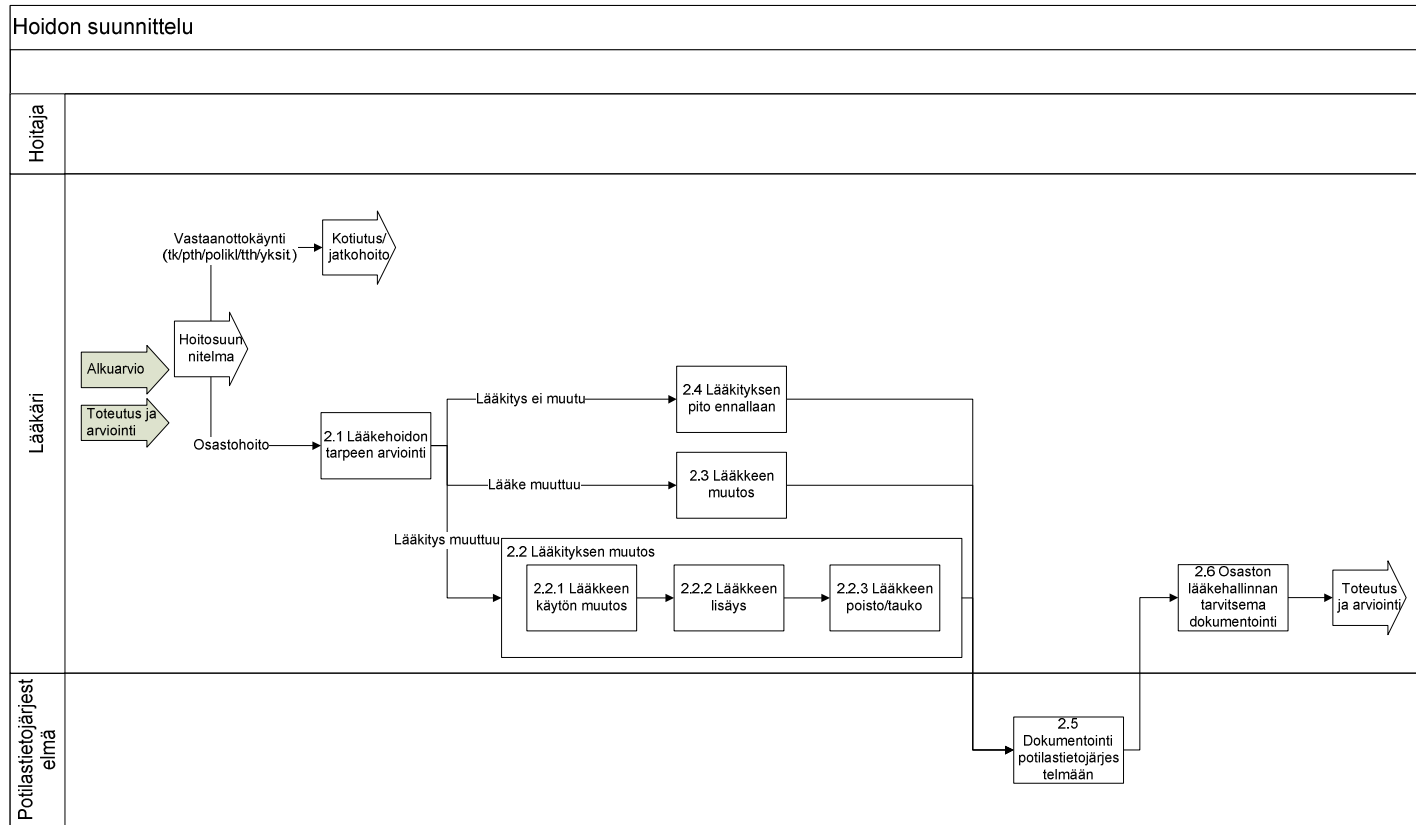


Liite 2: Lääkitystiedon prosessimallit [Nykänen ja muut, 2008]



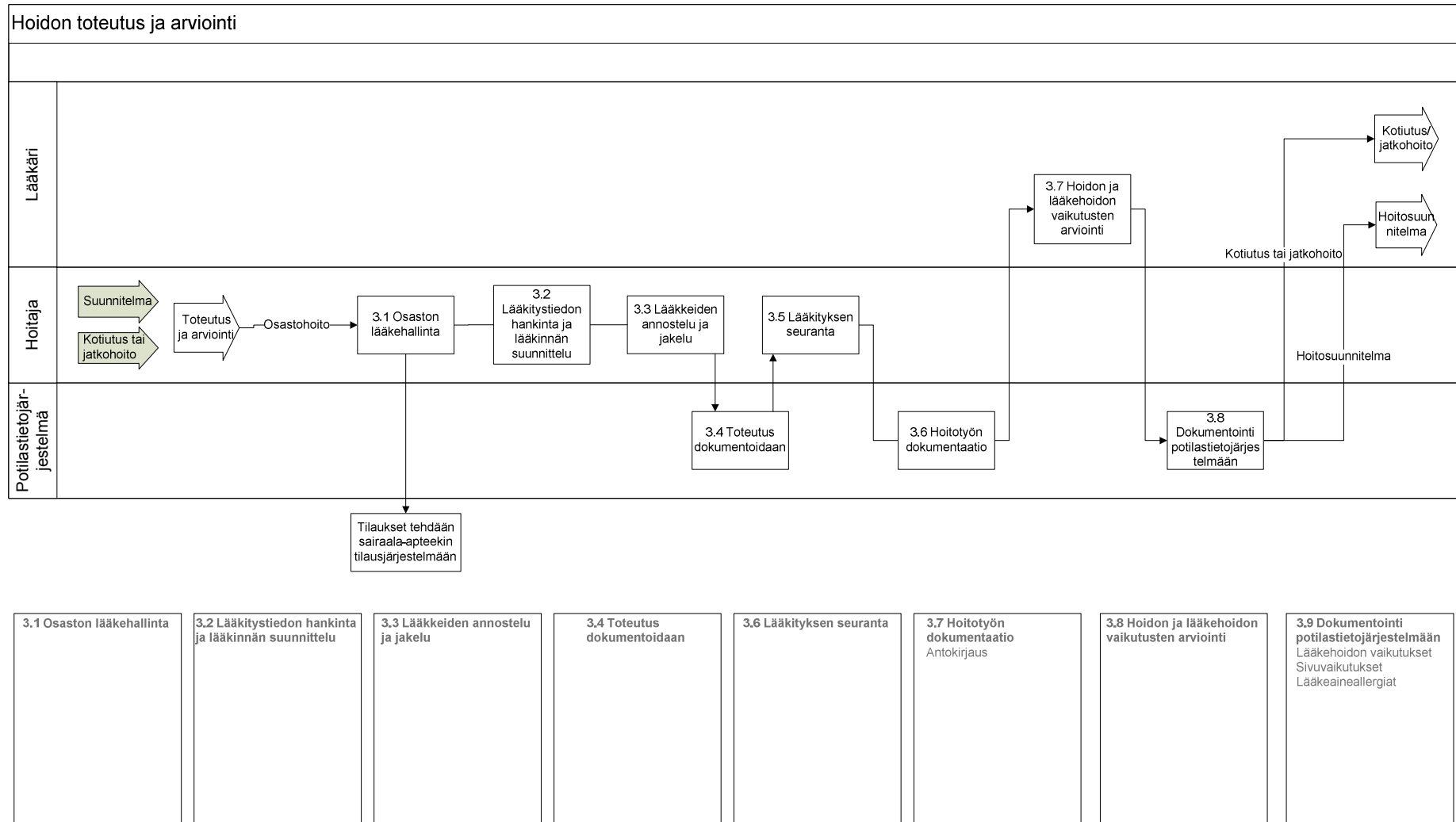
1.1 Potilaan tiedot potilastietojärjestelmästä	1.2 Alustava arvio potilaan tilasta	1.3 Potilaan lääkeyhityksen peruskartoitus	1.4 Hoidon tarvearvio	1.4.1 Nykytilan arviointi Nykytila: status ja anamneesi Laboratoriotutkimukset Muut tutkimukset <i>Lääkitys 1-3</i> + Tänäpä ja lähipäivinä otetut lääkkeet Aiempien suunnitelmien tavoitteiden toteutuminen	1.4.2 Sairaus-, hoito- ja lääkehityshistorian selvittäminen Diagnoosit Mitä hoidettu Miten hoidettu Missä hoidettu Hoidon määräjä Riskitiedot Toimintakyky <i>Lääkitys 4</i> Lääkitykseen vaikuttavat muut esitiedot	1.5 Nykytila-arvion dokumentointi potilastietojärjestelmään Diagnoosit Nykyinen toteutunut lääkehoito Lääkehoidon vaikutukset Lääkehoidon tarve	1.6 Hoidon tavoitteiden määrittely Tavoite: Sairaude parantaminen, oireiden helpottaminen, preventio
---	--	---	------------------------------	---	--	--	--

Kuva 1: Tulotilanne

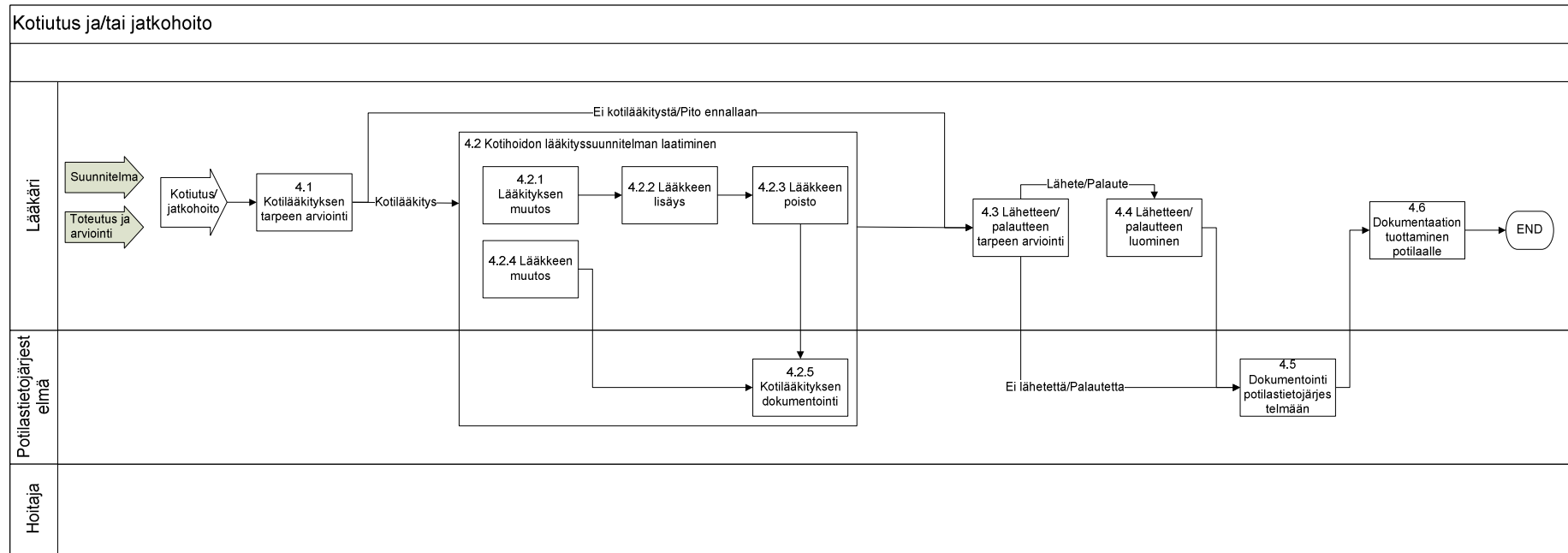


2.1 Lääkehoidon tarpeen arviointi Diagnoosit Lääkitys 1-3 Lääkkeiden vaikutukset	2.2.1 Lääkkeen käytön muutos <i>Lääke</i> Aiempi annostelu Uusi annostelu Muutoksen syy Muu ohjeistus (ruokavalio yms.) Käyttötarkoitus (indikaatio, diagnoosi tai hoidon tavoite)	2.2.2 Lääkkeen lisäys <i>Lääke</i> Annostelu Muu ohjeistus Lääkeryhmä (nark./antib.) Antokirjaus kyllä/ei Kertalääke kyllä/ei Muu ohjeistus (ruokavalio yms.) Käyttötarkoitus (indikaatio, diagnoosi tai hoidon tavoite)	2.2.3 Lääkkeen poisto/tauko	2.3 Lääkkeen muutos	2.4 Lääkityksen pito ennallaan	2.5 Dokumentointi potilastietojärjestelmään Muutokset lääkityksessä Muutosten perustelu	2.6 Osaston lääkehallinnan tarvitsema dokumentointi <i>Lääke</i> Lääkkeen annostelu Lääkeryhmä (nark./antib.) Antokirjaus kyllä/ei Kertalääke kyllä/ei Muu ohjeistus (ruokavalio yms.)
--	---	---	------------------------------------	----------------------------	---------------------------------------	--	---

Kuva 2: Hoidon suunnittelu



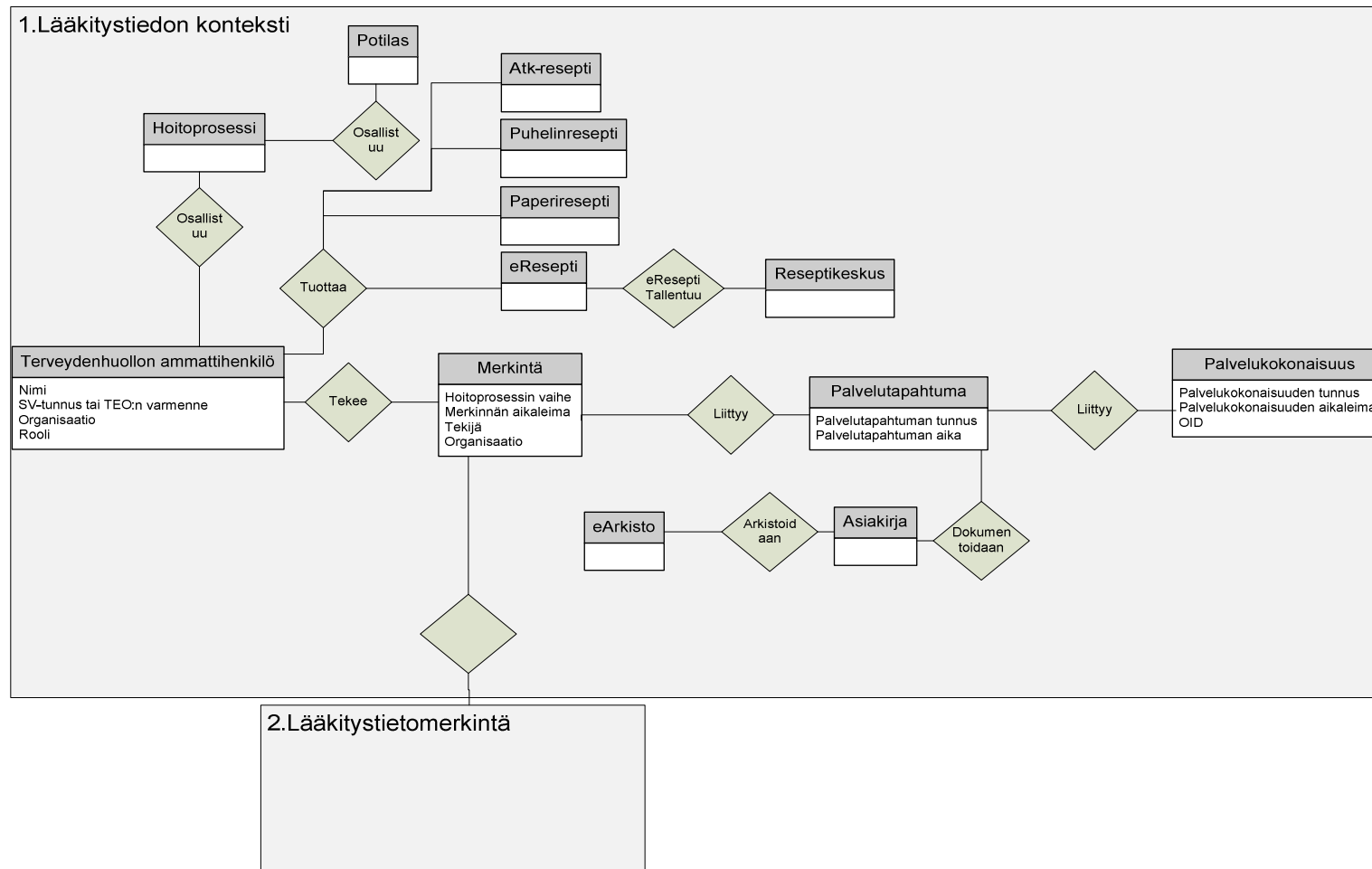
Kuva 3: Hoidon toteutus ja arviointi



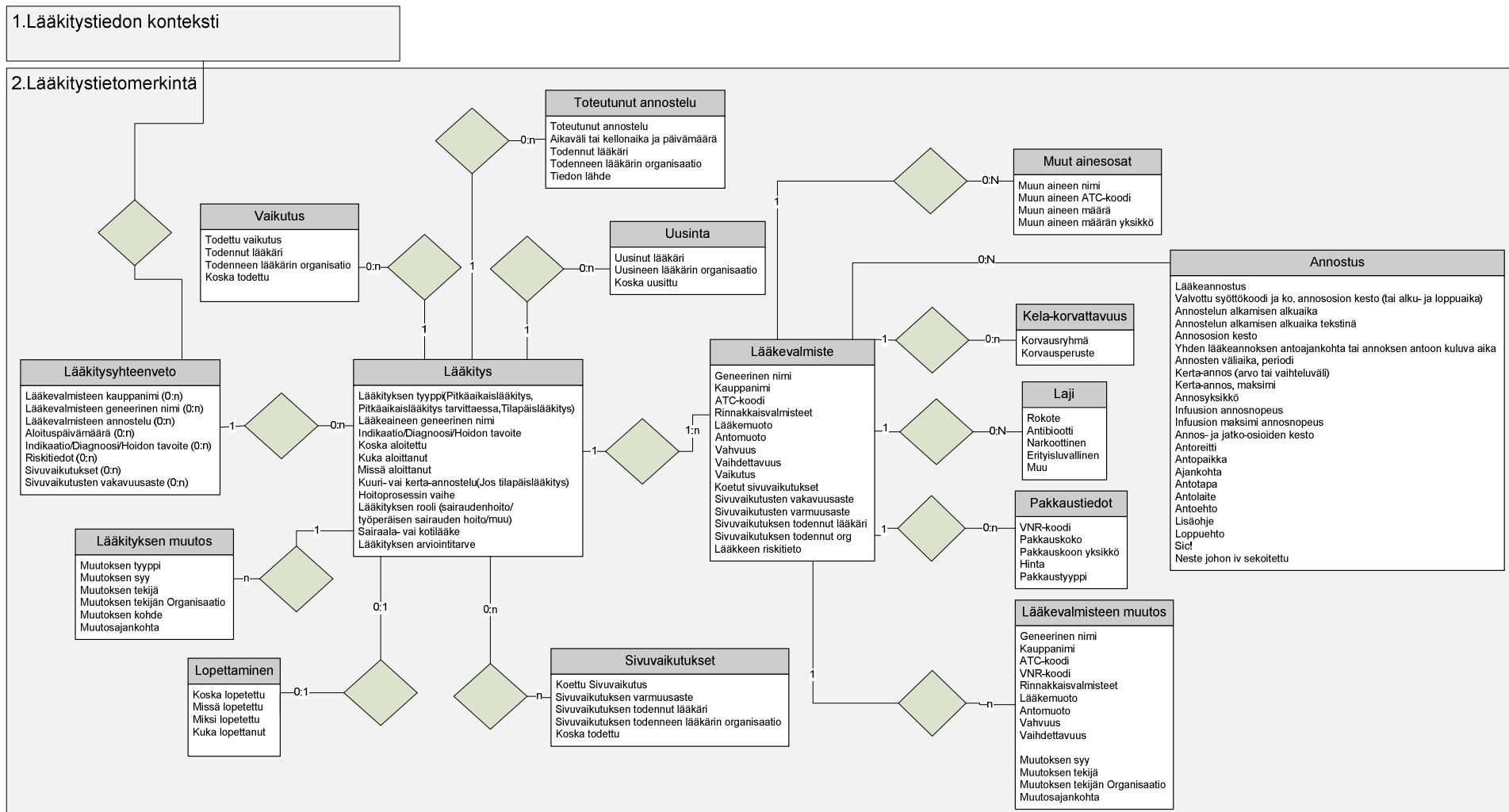
4.1 Kotilääkityksen tarpeen arviointi	4.2 Kotihoidon lääkitys suunnitelman laatiminen (lääkäri)	4.2.1 Lääkityksen muutos	4.2.2 Lääkkeen lisäys <i>Lääke</i> Lääkkeen hinta Lääkkeen SV-korvattavuus Antomuoto Annostelu Muu ohjeistus (ruokavalio yms.) Käyttötarkoitus (indikaatio, diagnoosi tai hoidon tavoite)	4.2.4 Lääkkeen muutos	4.2.5 Kotilääkityksen dokumentointi	4.3 Lähetteen/palautteen tarpeen arviointi	4.4 Lähetteen/palautteen luominen	4.5 Dokumentointi potilastietojärjestelmään	4.6 Dokumentaation tuottaminen potilaalle
---------------------------------------	---	--------------------------	--	-----------------------	-------------------------------------	--	-----------------------------------	---	---

Kuva 4: Kotiutus ja/tai jatkohoito

Liite 3: Lääkitystiedon tietomalli



Kuva 1: Lääkitystiedon konteksti



Kuva 2: Lääkitystietomerkinä

Liite 4: Medication D-MIM, POME_DM00000, v1.4.7 – 04 March 2008

Medication - D-MIM
POME_DM00000
v1.4.7 - 04 March 2008

Recent Change History

- v1.4.1 - 25 Jul 2006**
 - 1) Agreed D-MIM details, so that the DMM becomes a full aspect of the POME RIME
- See Annex 8 - 8 May 2006**
 - 1) Revised SPC/ACQ to Content consistency with SPL, & names were confusing
 - 2) Revised Medication to Part Medication/observation (per O. Characteristics)
 - 3) Revised Schemata to Role
 - 4) Associated Characteristics class with Part
 - 5) Associated Schemata with Medication
 - 6) Associated Schemata of Approval, Policy and Characteristics with Manufacture/Product
- v1.4.2 - 11 Sep 2006**
 - 1) Revised Characteristics and Policy class associated with Medication (left shadows in place, and MigrationProgram directly associated with Medication)
 - 2) Revised Schemata Characteristics to Medication/CharacteristicType
 - 3) Changed Policy to be ASD/AF rather than OAD/AD
 - 4) Changed Part to PackageMedicine from "This is the pack in which the medication is supplied to the patient" to "This is the pack in which the medicine is contained"
- 25 Oct 2006**
 - 1) Added relevant attributes to AdministrationGuideline
- v1.4.3 - 2 May 2007**
 - 1) Revised Administration CMEI from PDRM_00050 to CQC7_00000
- v1.4.4 - 19 Sep 2007**
 - 1) Revised some names
 - 2) Deleted player and source roles for country and agency of approval
- v1.4.5 - 15 Feb 2008**
 - 1) Moved observation goal to hang off Medication directly rather than via the shadow. Removed the shadow
 - 2) Revised independent/independent from AdministrationGuideline
 - 3) Moved all Agent/Entity and Observation resources on MIM to final
 - 4) Changed the CMEI ID for AdministrationGuideline
 - 5) Deleted class/Quantity from SubstantiveMedication
 - 6) Made registration/observation ObservationEvent/Class mandatory and default of FALSE
 - 7) Revised Form/Manufacture/Product
 - 8) Replaced independent/dependent with a Policy CMEI
 - 9) Fixed Medication/Characteristic/Class to RIME
 - 10) Changed Form, Specialization to Medication/Class to be optional 0..1
 - 11) Deleted Policy code as mandatory and changed the vocab domain to Medication/Characteristic/Type
 - 12) Made participation in ObservationGoal 0..1 rather than 1
 - 13) On SubstantiveAdministration corrected data type for subCode to 0..1 SET-CD (over 0..1)
 - 14) ObservationSubstantiveAdministrationType does not exist - replaced with domain Form/Characteristic/Class/Type
- v1.4.6 - 01 March 2008**
 - 1) Added entry point for Medication Lookup Response
 - 2) Changed Substance code from 1..1 to 0..1
 - 3) The following changes to vocabulary have been made:
 - Policy code - changed to ACQ/Code/Registration/Code from SPL
 - Substance code - changed to Observation/CodeType
 - Medicine code - Medicine is what is usually set to PackageMedicine code: EntryCode now Content/EntryCode
 - Manufacturer code - EntryCode - stopped - no obvious use nor suitable vocabulary
 - Approval code - AcCode changed to AdministrationApproval to match SPL
 - 4) Added MedicationDocument back
- v1.4.7 - 04 March 2008**
 - 1) Corrected artifact ID for Administration CMEI

TO DO

- Harmonization:**
1) Address appropriate domain (may already have been approved at harmonization)
SubstantiveMedication/Characteristic/CharacteristicType (in PackageMedicine)
Country (in Country)
CountryCode (in Country)
Registration/Policy/Code (in Policy)
AdministrationApproval (in Approval)
2) Add domain descriptions for:
Registration/Characteristic/Code (in PackageMedicine)
AdministrationApproval (in Approval)
3) Find better vocab domains for:
AdministrationApproval/Code - AcCode
- Release 1:**
1) Make sure use of SET-CD is correct
2) Medication/CharacteristicType or Characteristics appears to have incorrect definitions in RoleText
3) Delete AdministrationApproval: use set to XSD (if no set or instance mode to Release 2)
4) Review all vocabulary types for Codes
- Release 2:**
1) Look at changing name data type and cardinality to allow structured names
2) Add an EQUID role that allows memory retrieval for medicines and substances
3) Add to transfer readability to abstract and structure operators and recognition of base and sub amounts

v1.4.1 - 25 Jul 2006

- 1) Revised Administration CMEI from PDRM_00050 to CQC7_00000

v1.4.2 - 11 Sep 2006

- 1) Revised Characteristics and Policy class associated with Medication (left shadows in place, and MigrationProgram directly associated with Medication)
- 2) Revised Schemata Characteristics to Medication/CharacteristicType
- 3) Changed Policy to be ASD/AF rather than OAD/AD
- 4) Changed Part to PackageMedicine from "This is the pack in which the medication is supplied to the patient" to "This is the pack in which the medicine is contained"

25 Oct 2006

- 1) Added relevant attributes to AdministrationGuideline

v1.4.3 - 2 May 2007

- 1) Revised Administration CMEI from PDRM_00050 to CQC7_00000

v1.4.4 - 19 Sep 2007

- 1) Revised some names
- 2) Deleted player and source roles for country and agency of approval

v1.4.5 - 15 Feb 2008

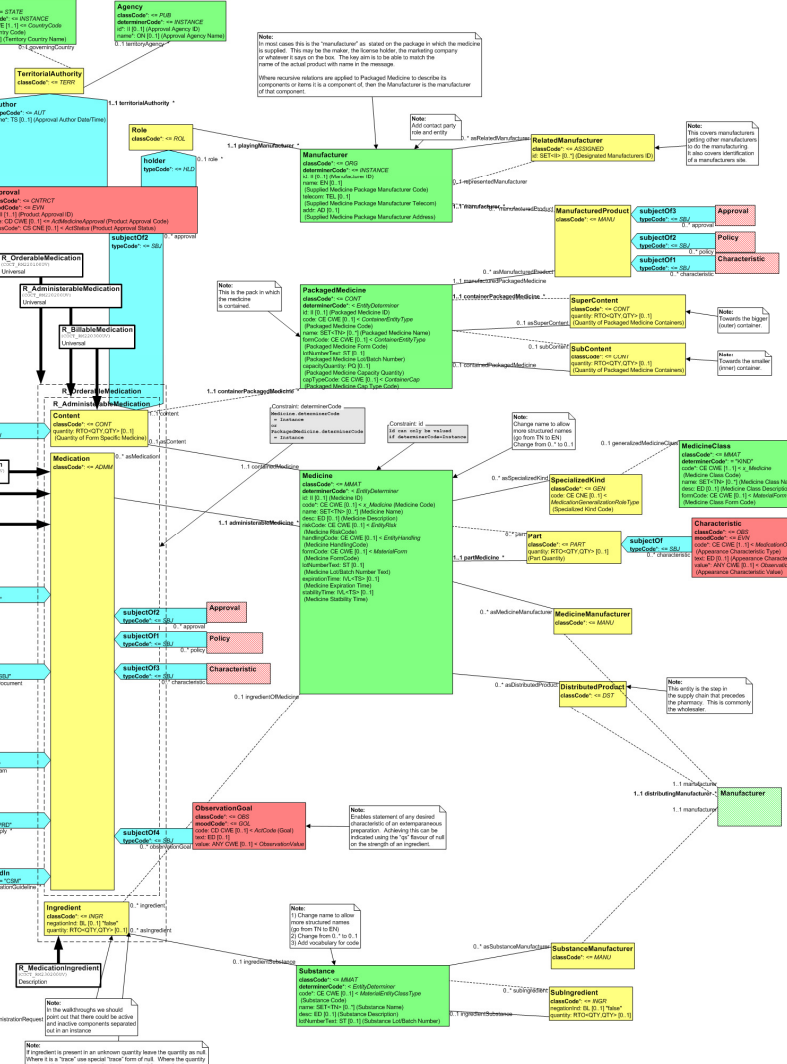
- 1) Moved observation goal to hang off Medication directly rather than via the shadow. Removed the shadow
- 2) Revised independent/independent from AdministrationGuideline
- 3) Moved all Agent/Entity and Observation resources on MIM to final
- 4) Changed the CMEI ID for AdministrationGuideline
- 5) Deleted class/Quantity from SubstantiveMedication
- 6) Made registration/observation ObservationEvent/Class mandatory and default of FALSE
- 7) Revised Form/Manufacture/Product
- 8) Replaced independent/dependent with a Policy CMEI
- 9) Fixed Medication/Characteristic/Class to RIME
- 10) Changed Form, Specialization to Medication/Class to be optional 0..1
- 11) Deleted Policy code as mandatory and changed the vocab domain to Medication/Characteristic/Type
- 12) Made participation in ObservationGoal 0..1 rather than 1
- 13) On SubstantiveAdministration corrected data type for subCode to 0..1 SET-CD (over 0..1)
- 14) ObservationSubstantiveAdministrationType does not exist - replaced with domain Form/Characteristic/Class/Type

v1.4.6 - 01 March 2008

- 1) Added entry point for Medication Lookup Response
- 2) Changed Substance code from 1..1 to 0..1
- 3) The following changes to vocabulary have been made:
 - Policy code - changed to ACQ/Code/Registration/Code from SPL
 - Substance code - changed to Observation/CodeType
 - Medicine code - Medicine is what is usually set to PackageMedicine code: EntryCode now Content/EntryCode
 - Manufacturer code - EntryCode - stopped - no obvious use nor suitable vocabulary
 - Approval code - AcCode changed to AdministrationApproval to match SPL
- 4) Added MedicationDocument back

v1.4.7 - 04 March 2008

- 1) Corrected artifact ID for Administration CMEI



Note: If an ingredient is present in an unknown quantity leave the quantity as null. Where a "max" or "spec" value is used, the quantity is to be adjusted to achieve a first value use "for" or "null".