

HL7 Finlandin suositus V3-viestien käyttöönottoon

**V 2.20.1**

**1.11.2018**

**OID: 1.2.246.777.11.2010.30**

Sisällysluettelo

[1. Käyttöönoton lähtötilanne 9](#_Toc200435589)

[2. Sanomanvälitys 10](#_Toc200435590)

[2.1 Perusteet 10](#_Toc200435591)

[2.2 Kuittaukset 12](#_Toc200435592)

[2.3 Viestien kehyskerrokset 18](#_Toc200435593)

[2.3.1 Siirtokehys – normaalit sanomat 18](#_Toc200435594)

[2.3.1.1 id (pakollinen, II) 20](#_Toc200435595)

[2.3.1.2 creationTime (pakollinen TS) 21](#_Toc200435596)

[2.3.1.3 securityText (vapaaehtoinen ST) 22](#_Toc200435597)

[2.3.1.4 versionCode (vapaaehtoinen CS) 22](#_Toc200435598)

[2.3.1.5 interactionId (pakollinen II) 22](#_Toc200435599)

[2.3.1.6 profileId (vapaaehtoinen SET(II) ) 22](#_Toc200435600)

[2.3.1.7 processingCode (pakollinen CS) 23](#_Toc200435601)

[2.3.1.8 processingModeCode (pakollinen CS) 23](#_Toc200435602)

[2.3.1.9 acceptAckCode (pakollinen, CS) 24](#_Toc200435603)

[2.3.1.10 sequenceNumber (vapaaehtoinen INT) 24](#_Toc200435604)

[2.3.1.11 attachmentText (vapaaehtoinen) 24](#_Toc200435605)

[2.3.1.12 receiver 25](#_Toc200435606)

[2.3.1.13 respondTo 26](#_Toc200435607)

[2.3.1.14 sender 26](#_Toc200435608)

[2.3.1.15 attentionLine (vapaaehtoinen) 27](#_Toc200435609)

[2.3.1.16 ControlActProcess 28](#_Toc200435610)

[2.3.2 Siirtokehys – kuittaussanomat 28](#_Toc200435611)

[2.3.2.1 acknowledgement – typeCode (pakollinen, CS) 30](#_Toc200435612)

[2.3.2.2 expectedSequenceNumber (vapaaehtoinen, INT) 31](#_Toc200435613)

[2.3.2.3 messageWaitingNumber (vapaaehtoinen, INT) 31](#_Toc200435614)

[2.3.2.4 messageWaitingPriorityCode (vapaaehtoinen, INT) 31](#_Toc200435615)

[2.3.2.5 targetMessage – id (pakollinen, II) 31](#_Toc200435616)

[2.3.2.6 acknowledgementDetail (vapaaehtoinen) 32](#_Toc200435617)

[2.3.3 Kontrollikehys - normaalisanomat 32](#_Toc200435618)

[2.3.3.1 classCode (pakollinen, CS) 34](#_Toc200435619)

[2.3.3.2 moodCode (pakollinen, CS) 34](#_Toc200435620)

[2.3.3.3 Id (SET(II) ) 34](#_Toc200435621)

[2.3.3.4 code (CD) 34](#_Toc200435622)

[2.3.3.5 text (ED) 35](#_Toc200435623)

[2.3.3.6 effectiveTime ( IVL(TS) ) 35](#_Toc200435624)

[2.3.3.7 priorityCode ( SET(CE) ) 35](#_Toc200435625)

[2.3.3.8 reasonCode ( SET(CE) ) 35](#_Toc200435626)

[2.3.3.9 languageCode ( CE) 35](#_Toc200435627)

[2.3.3.10 Osapuolet 35](#_Toc200435628)

[2.3.3.11 subject 36](#_Toc200435629)

[2.3.3.12 reasonOf 36](#_Toc200435630)

[2.3.4 Kontrollikehys – keskitetyt rekisterit 37](#_Toc200435631)

[2.4 Kyselysanomat 37](#_Toc200435632)

[2.4.1 Yleistä HL7v3:n kyselysanomista 37](#_Toc200435633)

[2.4.2 Yleinen kontrollikehyksen rakenne 40](#_Toc200435634)

[2.4.3 Query Specification-kontrollikehys 40](#_Toc200435635)

[2.4.4 Query Response/Acknowledgement-kontrollikehys 42](#_Toc200435636)

[2.4.5 Query Continuation/Cancel-kontrollikehys 44](#_Toc200435637)

[2.4.6 Esimerkki kyselyviestin ParameterList-rakenteesta 45](#_Toc200435638)

[2.4.7 Esimerkki kyselyvastauksen Subject-rakenteesta 45](#_Toc200435639)

[2.5 Batch-sanomat 48](#_Toc200435640)

[2.5.1 Batch siirtokehys 49](#_Toc200435641)

[2.5.1.1 Id (pakollinen II) 51](#_Toc200435642)

[2.5.1.2 creationTime (pakollinen TS) 51](#_Toc200435643)

[2.5.1.3 securityText (vapaaehtoinen ST) 51](#_Toc200435644)

[2.5.1.4 responseModeCode (pakollinen CS) 51](#_Toc200435645)

[2.5.1.5 versionCode (vapaaehtoinen CS) 51](#_Toc200435646)

[2.5.1.6 interactionId (pakollinen II) 51](#_Toc200435647)

[2.5.1.7 referenceControlId (required, voi olla null, II) 51](#_Toc200435648)

[2.5.1.8 name (vapaaehtoinen SC) 51](#_Toc200435649)

[2.5.1.9 batchComment (vapaaehtoinen, toistuva SET<ST>) 52](#_Toc200435650)

[2.5.1.10 transmissionQuantity (required INT) 52](#_Toc200435651)

[2.5.1.11 batchTotalNumber (vapaaehtoinen, toistuva SET<INT>) 52](#_Toc200435652)

[2.5.1.12 Message 52](#_Toc200435653)

[2.5.1.13 receiver (pakollinen) 53](#_Toc200435654)

[2.5.1.14 respondTo (vapaaehtoinen) 53](#_Toc200435655)

[2.5.1.15 sender (pakollinen) 53](#_Toc200435656)

[2.5.1.16 attentionLine (vapaaehtoinen) 53](#_Toc200435657)

[2.5.2 Response Batch siirtokehys 53](#_Toc200435658)

[2.5.2.1 message 54](#_Toc200435659)

[2.5.2.2 acknowledgement – typeCode (pakollinen, CS) 54](#_Toc200435660)

[2.5.2.3 acknowledgement – targetBatch – id (pakollinen, II) 54](#_Toc200435661)

[2.5.3 Batch Interaktiot 54](#_Toc200435662)

[2.5.3.1 Send Batch (MCCI\_IN200100UV) 55](#_Toc200435663)

[2.5.3.2 Send Response Batch (MCCI\_IN200101UV) 55](#_Toc200435664)

[2.5.3.3 Send message Adapter Batch Acknowledgement (MCCI\_IN200102UV) 55](#_Toc200435665)

[2.5.4 Suositeltu käyttötapa 56](#_Toc200435666)

[2.6 Vastaanottokuittaus - Accept Ack (MCCI\_IN000002UV01) Interaktio 56](#_Toc200435667)

[2.6.1 Vastaanottokuittaus synkronisessa/asynkronisessa liikenteessä 57](#_Toc200435668)

[2.7 Virheiden ilmoittaminen 58](#_Toc200435669)

[2.7.1 Virheiden ilmoittaminen siirtokehyksessä 58](#_Toc200435670)

[2.7.1.1 Vastaanottokuittaus 58](#_Toc200435671)

[2.7.1.2 Sovellustason kuittaus 60](#_Toc200435672)

[2.7.2 Sovellustason virheet kontrollikehyksessä 60](#_Toc200435673)

[3. Tiedonsiirto 63](#_Toc200435674)

[3.1 Yleiset periaatteet 63](#_Toc200435675)

[3.1.1 perusperiaatteet 63](#_Toc200435676)

[3.1.2 Siirron luotettavuus 65](#_Toc200435677)

[3.1.2.1 Siirtoprotokollataso – commit ack 65](#_Toc200435678)

[3.1.2.2 Sovellustason adapterit ja vastaanottokuittaus 65](#_Toc200435679)

[3.1.2.3 Gateway 66](#_Toc200435680)

[3.1.2.4 Bridge - silta 67](#_Toc200435681)

[3.1.2.5 Reititin (intermediary, router) 67](#_Toc200435682)

[3.1.3 Osoitteistus 68](#_Toc200435683)

[3.1.4 Yhteenvetokuva 68](#_Toc200435684)

[3.2 Web services tiedonsiirto 70](#_Toc200435685)

[3.2.1 Web services yleiskuvaus 70](#_Toc200435686)

[3.2.2 HL7:n web services profiilit 71](#_Toc200435687)

[3.2.3 Perusprofiili (HL7 Web services basic profile, HL7WSP) 71](#_Toc200435688)

[3.2.3.1 SOAP-body 72](#_Toc200435689)

[3.2.3.2 WSDL-kuvaukset 73](#_Toc200435690)

[3.2.3.3 Synkroninen Web Services siirto 79](#_Toc200435691)

[3.2.3.4 Yhteensopivuus 81](#_Toc200435692)

[3.2.4 Osoitteistusprofiliili (HL7WSAP) 82](#_Toc200435693)

[3.2.5 Tietoturvaprofiilli (HL7WSSP) 86](#_Toc200435694)

[3.2.6 Luotettava sanomanvälityksen profiili (HL7RMWSP) 87](#_Toc200435695)

[3.3 Määrityksistä käytettävät versiot 88](#_Toc200435696)

[3.4 Http-siirto 88](#_Toc200435697)

[4. Semanttisen sisällön implementointi 89](#_Toc200435698)

[4.1 HL7-mallinnuksen perusperiaatteet 89](#_Toc200435699)

[4.2 Vocabularyn periaatteet 92](#_Toc200435700)

[4.2.1 Vocabulary domain 93](#_Toc200435701)

[4.2.2 Value Set 93](#_Toc200435702)

[4.2.3 Code system 93](#_Toc200435703)

[4.2.4 OID 94](#_Toc200435704)

[4.3 Lähtökohta 94](#_Toc200435705)

[4.4 Perusperiaate 95](#_Toc200435706)

[4.5 HL7-domainit 95](#_Toc200435707)

[4.6 Artifaktien nimeäminen 96](#_Toc200435708)

[4.7 Implementointioppaan muoto 97](#_Toc200435709)

[4.8 Sovellusalueen yleiskuvaus 98](#_Toc200435710)

[4.9 Interaktiopohjainen lähestymistapa 98](#_Toc200435711)

[4.10 Muita ohjeita 102](#_Toc200435712)

[5. Jatkokehitys 102](#_Toc200435713)

[6. Esimerkki 102](#_Toc200435714)

LIITETIEDOSTOT:

MCCI\_DM000000.png Siirtokehyksen D-MIM

MCCI\_RM000100.png Perussanoman lähetyksen R-MIM

MCCI\_HD000100UV01.htm Perussanoman lähetys taulukkomuodossa

MCCI\_RM000300.png Sovellustason vastaussanoman R-MIM

MCCI\_HD000300UV01.htm Sovellustason vastaussanoma taulukkomuodossa

MCCI\_RM000200.png Vastaanottokuittauksen R-MIM

MCCI\_HD000200UV01.htm Vastaanottokuittaus taulukkomuodossa

MCAI\_DM700200.png Kontrollikehyksen D-MIM

MCAI\_RM700200UV01.png Kontrollikehyksen R-MIM

MCAI\_HD700200UV01.htm Kontrollikehys taulukkomuodossa

MCAI\_RM900000.png DetectedIssueEvent R-MIM

MCAI\_HD900000.htm DetectedIssueEvent taulukkomuodossa

MCCI\_RM200100.png Batch wrapper R-MIM

MCCI\_RM200101.png Response Batch wrapper R-MIM

Wslyhe.doc WS-tiedonsiirron termejä

v3esimerkki.xml Esimerkkitiedosto siirtovalmiista aineistoista

MCCI\_IN000002UV01\_CA\_esim.xml Positiivinen vastaanottokuittaus esimerkki

MCCI\_IN000002UV01\_CE\_esim.xml Virhe vastaanottokuittaus esimerkki

MCCI\_IN000002UV01\_CR\_esim.xml Reject vastaanottokuittaus esimerkki

PRSC\_AR030101.wsdl wsdl esimerkkitiedosto (SerAPI Ajanvaraussanomat aineistosta)

MCCI\_MT000100UV01.xsd Siirtokehys – normaalisanomat scheema

MCCI\_MT000200UV01.xsd Siirtokehys – vastaanottokuittaus scheema

MCCI\_MT000300UV01.xsd Siirtokehys – sovellustason kuittaus scheema

MCAI\_MT700200UV01.xsd Kontrollikehys – normaalisanomat scheema

MCCI\_IN200100UV.xsd Batch wrapper interaktion esimerkkischeema

MCCI\_IN200101UV.xsd Response Batch interaktion esimerkkischeema

MCCI\_IN002200UV.xsd Batch kuittaus interaktion esimerkkischeema

**Versiohistoria**

|  |  |  |  |
| --- | --- | --- | --- |
| Versio | Pvm | Tekijät | Selite |
| 0.3 | 13.12.2005 | Timo Tarhonen/Tietotarha  Esko Eloranta/Tietotarha | Ensimmäinen versio SerAPI:n asiantuntijoille tutkittavaksi |
| 0.99 | 11.1.2006 | Timo Tarhonen, Esko Eloranta/Tietotarha | Julkaisuversio.  Kirjoitettu hiukan lisää control actistä ja lisätty esimerkki (luku 6) ja http-luku 3.3. |
| 0.999 | 22.1.2006 | Marko Sormunen  Jari Porrasmaa  Kuopion yliopisto | Lisätty luku 2.4 kyselysanomat ja toimitettu kommentteja dokumentin muihin osiin |
| 0.9991 | 31.1.2006 | Huomioitu Kuopion yliopiston kommentit | Julkaisuversio |
| 1.0 | 28.2.2006 | Timo Tarhonen/Tietotarha | Lopullinen version vuoden 2006 pakettiin. Huomioitu open CDA-työryhmän kommentit ja otettu huomioon Ari Vähä-Erkkilän /Prime kommentit. |
| 1.1 | 7.9.2007 | Timo Tarhonen Esko Eloranta / Tietotarha | Sanomarakenteet V3 2006 standardin mukaisiksi. Lisätty Batch kappale, Accept Ack interaktio, virheenkäsittely, synkroninen web services siirto, paikalliset lisäykset reference elementillä ja vocabularyn periaatteet |
| 2.0 | 18.1.2008 | Timo Tarhonen Esko Eloranta / Tietotarha | Kuopion yliopiston kommentit lisätty.  WS-addressing reference elementin virhe korjattu ja hl7fi elementtejä tarkennettu.  Tarkennettu siirtokehyksen id:tä, kontrollikehyksen osapuolten organisaation esittämistä ja relatesTo elementtiä webServices siirrossa. |
| 2.01 | 31.3.2008 | Timo Tarhonen Esko Eloranta / Tietotarha | Batch osiota päivitetty, batch esimerkkischeemat lisätty |
| 2.02 | 5.6.2008 | Timo Tarhonen Esko Eloranta / Tietotarha | Tehty äänestysversio lausuntokierroksen kommenttien perusteella.  Lisäselitystä seuraaviin kohtiin: 2.2 http-koodit korjattu (myös 3.1.2.1), 2.3.3.4 triggerin code, 2.3.1.4 Version code, 2.3.1.5 InteractionId, 2.3.1.9 acceptAckCode, 2.3.3.2 moodCode, 2.5.1.4 responseModeCode, 2.5.4 batchin suositeltu käyttötapa, 3.2.3.2 @wsa:Action käytön lievennys,3.2.4 paikalliset header lisäykset DdocumentType ja tableOfContents poistettu |
| 2.10 | 4.12.2008 |  | Hyväksytty HL7 Finlandin äänestyskierroksella 8/2008 ja teknisessä komiteassa 23.9.2008 |
| 2.11 | 15.9.2010 | Pekka Rinne, Marko Jalonen / Kela | Lukuun 3.2.4 lisätty HL7 teknisessä komiteassa 24.8.2010 päätetty linjaus ReplyTo-elementin käytön poistamisesta SOAP-headerien WS-Addressing kehyksistä synkronisessa Web Services –liikenteessä + poistettu [reply endpoint] –elementin pakollisuus ko. luvun taulukosta sekä lisätty huomautus KanTa-palveluiden WS-addressing –linjauksista. Lukuihin 2.3.1.12 (receiver) ja 2.3.1.14 (sender) lisätty viittaukset KanTa-palvelujen osapuolitunnusohjeistukseen |
| 2.20 | 14.10.2010 | Pekka Rinne / Kela | Dokumentin OID-tunnus ja versionumero päivitetty julkaisuversiota varten. |
| 2.20.1 | 1.11.2018 | Kela  Kela | Huom. dokumentin OID-tunnusta ei ole muutettu, vain versionumeroa nostettu (minor-muutos, nämä muutokset ovat olleet voimassa jo ennen näitä täsmennyksiä).  Kpl 2.3.1.1 täsmennetty id:n käyttöä Kanta-palveluiden HL7 V3 sanomissa (UUID ei sallittu).  Kpl 3.2.4: Täsmennetty wsa:n käyttöä Kanta-palvelujen HL7 V3 sanomaliikenteessä. |

Lukijalle:

Dokumentti on lähinnä referenssiopas HL7v3-viestinnän kanssa työskenteleville henkilöille, se ei itsessään kunnolla sovellu opiskeluun, jos henkilöllä ei ole ennakkoon juurikaan tietoa HL7v3-viestinnästä.

Tämä opas pyrki määrittelemään sanomien kuorikerrokset ja sanomien siirrossa käytettävän (web services) tekniikan siten, että se olisi sovellettavissa. Todellisten implementointien yhteydessä määrityksiä joudutaan tarkentamaan.

Lisäksi tämä opas pyrki antamaan perusohjeita V3 messaging – implementointioppaiden tekoon ja HL7 V3 ymmärtämiseen.

Tässä dokumentissa esitetty sanomanvälitys ja tiedonsiirto perustuvat (HL7 International) V3 2006 viralliseen standardiin ja May 2007 äänestyspakettiin. May 2007-äänestyspaketista on otettu batch-määrittelyt. Sanomanvälityksestä on tulossa versio R2 arviolta kesällä 2008, tämä määritys on siis R1-versiota.

# Käyttöönoton lähtötilanne

potilaskertomustietojen

siirto CDA R2

Yleiset palvelut

(API-tyyppiset)

hoidon järjestämiseen ja toteutukseen liittyvä sanomanvälitys (labra, potilashallinto, ajanvaraus jne.) V2>V3

Kuva 1

Käyttöönoton lähtötilanne

Kansallisessa terveysprojektissa on keskitytty potilaskertomustietojen siirtomääritykseen HL7 CDA R2 – standardin avulla. Yleisiä sovellusrajapintapalveluja (mm. kontekstinhallinta, koodistojen hallinta ja käyttäjien ja käyttöoikeuksien hallinta) on määritelty Kuopion yliopiston hankkeissa ja sittemmin ne on hyväksytetty kansallisiksi HL7-määrityksiksi. Hoidon järjestämiseen ja toteutukseen liittyvä sanomanvälitys on perustunut HL7 versioon 2.3 ja se on toiminut hyvin.

Useissa maissa on kuitenkin aloitettu HL7 version 3 käyttö sanomanvälityksessä ja ennen pitkää se on natiivirajapinta uusissa tuotteissa, silloin kun on kyse ”tuotannonohjauksen tapaisesta” sanomaintegraatiosta. Tämän vuoksi Suomessakin on korkea aika aloittaa HL7 V3 sanomanvälitykseen liittyvien implementointioppaiden teko, jotta pysyisimme oikealla kehityspolulla. Versio 3:een on siirrytty joissain paikoissa viranomaismääräyksinä (esim NHS). Tyypillisin tapa on kuitenkin ottaa V3 messaging käyttöön sovellusalueilla, joissa se ei ole ollut ollenkaan tai laajassa käytössä. Käyttöönotto on ollut kuitenkin odotettua hitaampaa, koska HL7 V3 sanomanvälityksen implementointi on koettu vaikeaksi ja se vaatii monien uusien asioiden opiskelua. Suomessa V3:n käyttö on aloitettu ajanvarauksista. Kansallistetut ajanvarausmääritykset on hyväksytty äänestyksessä kesällä 2007. Samoin on hyväksytty V3-kuolinilmoitussanoma. Laajaan käyttöön V3-siirto tulee E-reseptin käyttöönoton myötä vuonna 2008.

Seuraavissa kappaleissa selitetään HL7 V3 tiedonsiirtoon liittyvät keskeiset rakenteet, sanoman kuorikerrokset eli kehykset ja siirto web services tekniikalla. Lopuksi annetaan ohjeita HL7 V3 tietosisällön paikallistamisoppaiden tekemiseen ja samalla V3 tietosisällön ymmärtämiseen.

# Sanomanvälitys

## Perusteet

Sanomanvälitys on HL7-RIM:stä johdettu yksi sovellusalue (domain). Sovellusalueen tunnus on MC. Sovellusalue jakautuu kahteen alidomainiin MCCI (siirtokehys) ja MCAI (kontrollikehys) sanoman kuorikerrosten mukaisesti. Siirtokehyksen sovellusalueen D-MIM (domain information model) on liitetiedostossa MCCI\_DM000000.png.

HL7-sanomat esitetään siirron aikana XML-muodossa. XML-schemat saadaan julkaisujärjestelmän työkaluilla ”nappia painamalla” sen jälkeen, kun semanttiset määritykset on saatu tehtyä. On tarkoitus, että käytämme Suomessa mahdollisimman pitkälle kansainvälisiä sanomamäärityksiä. Omia muutoksia tehdään vain pakottavassa tarpeessa.

Siirtotavasta riippumatta sanomilla on aina sama rakenne. Puhutaan HL7 composite messagesta. Siirrettävän sanoman uloin kerros on siirtokehys (transmission wrapper) ja sen alla on kontrollikehys (control act wrapper). Varsinainen payloadin sisältävä sanomamääritys on ripustettu kontrollikehykseen. Käytettävä siirtokehys ja kontrollikehys riippuvat interaktiosta.

Kuva 2

HL7-sanoman rakenne

(HL7 composite message)

Payload – varsinainen sanoma

(ei vastaanottokuittauksissa eikä kaikissa sovellustason kuittauksissa)

Kontrollikehys (ei vastaanottokuittauksissa)

Siirtokehys (aina)

Edellisessä kuvassa rakenne on kuvattu XML-tekniikan kannalta. Semanttisessa kerroskuvassa kerrokset olisivat toisin päin.

Sanomanvälitystä on periaatteessa kolmentyyppistä:

* Perussanomat
* Kyselyt ja niiden vastaukset
* Keskitettyjen rekistereiden ylläpito

Siirtokehys sisältää tietoja, joiden perusteella sanomanvälitysjärjestelmä saa lähetettyä sanoman oikealle vastaanottajalle. Se sisältää myös tietoja, jotka ohjaavat sanoman käsittelyä vastaanottopäässä (interaktio). Siirtokehyksessä mm. pyydetään vastaanottokuittausta (= adapterikuittaus) ja vastaanottokuittauksessa kuittaustiedot ovat siirtokehyksessä.

Kontrollikehys sisältää varsinaisen payloadin käsittelyä ohjaavia hallinnollisia tietoja, kuten kuka, mitä, missä ja milloin. Se kertoo, miksi sanoma on yleensä lähetetty (trigger event code). Tämäntyyppiset ”prosessinohjauksen” tiedot on nostettu omalle tasolleen, koska se helpottaa varsinaisten sanomien tietosisällön tekemistä. Kontrollikehys on tärkeä etenkin dynaamisen mallin rakentamisessa. Edellä esitetty sanomaliikenteen jako kolmeen kategoriaan vaikuttaa etenkin kontrollikehykseen, ei niinkään siirtokehykseen. Itse asiassa kontrollikehyksen määritykset muodostavat oman domaininsa MCAI.

Kontrollikehykseen ripustettu varsinainen tietosisältö (payload) riippuu määritellystä interaktiosta. Kullekin interaktiolle on oma schemansa. Varsinainen sanoma eli payload ei voi koskaan olla ilman kontrollikehystä, sillä kontrollikehys ja varsinainen sanoma muodostavat semanttisen kokonaisuuden. Kontrollikehys kertoo, että tee tietty asia tällä sisällöllä eli varsinaisella sanomalla.

Tässä dokumentissa ei ole käsitelty keskitettyjen rekistereiden ylläpitosanomia, vaan niistä tehdään myöhemmin omat implementointioppaansa.

Yksi mahdollisuus sanomien perillemenon takaamiseen oikeassa järjestyksessä on sequence number protokollan käyttö. Se toteuttaa myös duplikaattieston. Sanoman id:hen järjestyksen takaajana ei pitäisi standardin mukaan luottaa. Uusissa siirtoprotokollatason vaatimuksissa kuitenkin vaaditaan siirtoprotokollatasolta järjestyksen säilyttämistä. Osa siirtoprotokollatason toteutuksista myös hoitaa duplikaattieston. Tässä dokumentissa suositellaan kuitenkin message.id:tä duplikaattiestoon ja järjestyksen tarkistamiseen, koska message.id muodostetaan OID:in avulla vuosisarjoittain. (Periaatteessa OID takaa vain id:n yksilöllisyyden, OID:n käyttö järjestyksen tarkistamiseen vaatii paikallisen sopimisen.) Sequence number protokollan toteutus on hankalaa ja tuo sinällään jopa ylläpidon aikaisia ongelmia. Sequence number – protokollaan liittyvät tiedot on sijoitettu siirtokehykseen. HL7 Finland suosittelee Sequence number – protokollan sijasta käytettäväksi WS-ReliableMessaging –määritystä, jos pelkän message.id:n käyttö ei riitä. WS-ReliableMessaging –perustuu siirtoprotokollatasolla SOAP-headereihin ja se on määritelty dokumentissa ”Transport Specification - Web Services Profile, Release 2” (May 2007 HL7-äänestyspaketti).

Sanomien välitykselle on määritelty kaksi perusinteraktiota, joita eri domainien tekniset komiteat käyttävät varsinaisten implementoitavien interaktioiden pohjana. Hierarkiset sanomakuvaukset ovat:

MCCI\_HD000100UV01

perussanoman lähetys (R-MIM liitetiedostossa MCCI\_RM000100.png)

MCCI\_HD000300UV01

sovellustason vastaussanoma (R-MIM liitetiedostossa MCCI\_RM000300.png)

Ainoa tässä dokumentissa kuvattu suoraan implementoitava interaktio on vastaanottokuittaus MCCI\_IN000002UV01, jonka R-MIM on liitetiedostossa MCCI\_RM000200.png ja hierarkinen sanomakuvaus tiedostossa MCCI\_HD000200UV01.htm. Interaktio on kuvattu tämän dokumentin kohdassa 2.6.

## Kuittaukset

HL7:ssa on kolmen tason kuittauksia:

* Sovellustason vastaussanoma (kuittaus), jonka lähettäminen on määritelty sovellustason interaktiossa interaktion receiver responsibilities osiossa. Sovellustason kuittausta ei voi sanomassa erikseen pyytää vaan se on määritelty interaktion kuvauksessa interaktiona, jonka lähetettävä sanoma käynnistää. Yleensä receiver responsibilities osiossa on kaksi vaihtoehtoista interaktiota sovellustason kuittaukseksi esim. pyynnön hyväksyminen/hylkäys. Sovellustason kuittauksena voi myös toimia kyselytyyppisen sanoman vastaussanoma. Sanomanvälityksen kannalta sovellustason kuittaus on lähes normaali sovellustason sanoma, lisänä on acknowlegement luokka siirtokehyksessä. Siirtokehys on MCCI\_RM000300. Sovellustason kuittaukset lähetetään ensisijaisesti osoitteeseen, joka on ilmoitettu alkuperäisen sanoman siirtokehyksen elementissä respondTo/entityRsp/id, jos se on sanomassa mukana. Jos respondTo elementtiä ei ole käytetään sender-tietoa (sender/device/id), joka on pakollinen. Käytännössä respondTo elementti ei esiinny sanomissa jos se olisi sama kuin sender ja kuittaukset lähetetään sender elementin ilmoittamaan osoitteeseen. Sovellustason tarkistuksissa ja kuittauksissa on kyse semanttisen tietosisällön tarkistamisesta. Sovellustason vastaussanoma (interaktio) on normaali interaktio siinä mielessä että sen receiver responsibilities osiossa voi olla interaktioita, joilla siihen on vastattava.
* Vastaanottokuittaus (Accept Ack), jonka sovellusadapteri lähettää. Sovellusadapteri on välikerros (ohjelma), joka erottaa tietoliikenteen sovelluksesta. Sovellusadapteri ei ole pakollinen, vaan varsinainen sovellus voi hoitaa myös sovellusadapterin tehtävät. Vastaanottokuittaus Accept Ack muodostaa interaktion MCCI\_IN000002UV01. Vastaanottokuittausta pyydetään siirtokehyksessä acceptAckCode attribuutissa. Negatiivinen vastaanottokuittaus voi tulla myös sillalta (bridge), jos sanomanvälitys epäonnistuu. Vastaanottokuittauksessa ei ole kontrollikehystä eikä erillistä payloadia, vaan pelkkä siirtokehys (MCCI\_RM000200). Kuittaustiedot sijoitetaan siirtokehyksen luokkaan acknowledgement. Jokaisen vastaanottavan sovelluksen pitää osata lähettää vastaanottokuittaus (pyydettäessä) ja jokaisen lähettävän sovelluksen pitää pystyä vastaanottamaan vastaanottokuittaus. Vastaanottokuittaukset lähetetään aina osoitteeseen, joka on ilmoitettu alkuperäisen sanoman siirtokehyksen elementissä sender/device/id. Vastaanottokuittauksessa on kyse syntaksin ja teknisten asioiden tarkistamisesta. Positiivinen vastaanottokuittaus tarkoittaa, että viesti menee sovellustason käsittelyyn ja sen voi lähetysmielessä unohtaa. Luvussa 2.6 on määritelty tarkemmin mitä lähettäjä voi päätellä vastaanottokuittauksesta. Positiivinen vastaanottokuittaus lähetetään, jos mm. seuraavat ehdot toteutuvat:
  + vastaanottaja on tallentanut viestin onnistuneesti
  + viesti on tullut oikealle vastaanottajalle
  + viesti on XML-mielessä hyvin muodostettu (well formed)
  + viesti noudattaa schemaa (valid)
* tiedonsiirtokuittaus commit ack, joka liittyy siirtoprotokollaan eikä siis HL7:aan. Sen avulla toteutetaan luotettava sanomanvälitys. Commit ack lähetetään sen jälkeen, kun sanoma on vastaanotettu ja talletettu turvallisesti. Sisältö voi olla mitä hyvänsä (sovellustason normaali sanoma tai jokin ylemmän tason kuittaussanoma).



Kuva 3 Sanomaliikenteen peruskaavio

Lähde: Transmission Infrastructure – Generic message transmission topic, ballot sep 2005

Kuittauksia sovellusadapterin kannalta on käsitelty luvussa 3.1.2.2.

(Adapteritason) vastaanottokuittauksen (Accept Ack) interaktio on MCCI\_IN000002UV01, joka on kuvattu kohdassa 2.6.

Sovellustasolle on määritelty HL7:aan perustuvat kuittaukset (Kanta-palvelut), joten viitetietokantaympäristön CDA-tiedonsiirrossa käytetty sovellustason kuittaussanoma ei ole enää tarpeen. Viitetietokannoissa CDA-tiedonsiirrossa käytetään kuitenkin edelleen avoimet rajapinnat-määrityksiä.



Kuva 4 Wrapperit kuittauksissa ja virheiden paikat



Kuva 5.1 sovellustason kuittaus ja commit ack

(Mukaeltu copyright Ringholm GmbH 2005)

Kuvassa 5.1 sovellus (sender) antaa WS-adapterille sanoman siirrettäväksi. Siirtoprotokollatasolla lähettäjä (source) lähettää sanoman (1) vastaanottajalle (destination). Destination ”lähettää” sourcelle commit ack:in (2), joka tässä tapauksessa on 2XX http-koodi,esim. 202 message accepted. (Negatiivinen commit ack olisi ollut jokin muu kuin 2XX http-koodi tai sitten SOAP fault-sanoma). Destination antaa sanoman (3) vastaanottopäässä sovellukselle (receiver). Sanomassa onkin vikaa ja receiver lähettää negatiivisen sovellustason kuittauksen (vastaussanoman) (4). Kuittaustiedot sijoitetaan kontrollikehykseen (controlActWrapper) elementtiin detectedIssueEvent (tässä on kyseessä sovellustason virhe, ei vastaanottokuittaus eli Accept Ack). Nyt siirtotasolla roolit vaihtuvat ja siirto tapahtuu kuten toiseenkin suuntaan. Sovellustason kuittaus kuitataan siirtoprotokollatasolla commit-ackillä (5). Tässä kuvassa ei ole mukana adapteritason vastaanottokuittauksia (Accept Ack).

Kuva 5.2 Vastaanottokuittaus (Accept Ack)

(Mukaeltu copyright Ringholm GmbH 2005)

Kuvassa 5.2 sovellus (sender) antaa WS-adapterille sanoman siirrettäväksi. Siirtoprotokollatasolla lähettäjä (source) lähettää sanoman (1) vastaanottajalle (destination). Destination ”lähettää” sourcelle commit acki:in (2), joka tässä tapauksessa on 2XX http-koodi, esim 200 tai 202. (Negatiivinen commit ack olisi ollut jokin muu kuin 2XX http-koodi tai sitten SOAP fault-sanoma). Destination antaa sanoman (3) vastaanottopäässä sovellusadapterille (receiver). Sanomassa onkin kelvoton (ei ole hyvin muodostettua) XML-muoto ja sovellustason adapteri ei pysty toimittamaan sanomaa sovellukselle sovellusrajapinnan kautta. Adapteri (receiver) lähettää negatiivisen vastaanottokuittauksen (4). Kuittaustiedot sijoitetaan siirtokehykseen (transmission wrapper) elementtiin acknowledgement. Nyt siirtotasolla roolit vaihtuvat ja siirto tapahtuu kuten toiseenkin suuntaan. Vastaanottokuittaus kuitataan siirtoprotokollatasolla commit-ackillä (5). Myös positiivinen vastaanottokuittaus voidaan lähettää jos tulevan sanoman siirtokehyksen acceptAckCode elementissä sitä pyydetään.

Tekniset kuittaustiedot (etenkin vastaanottokuittaus kuvassa 5.2) sijoitetaan siis siirtokehykseen elementtiin acknowledgement. Sovellustason kuittaus/vastaussanomassa virhetiedot (sovellustason vastaus kuvassa 5.1) sijoitetaan kontrollikehykseen elementtiin detectedIssueEvent. On huomattava, että sovellustason vastaussanomassakin tarvitaan acknowledgement-luokkaa sovellustason kuittauskoodin välittämiseen, acknowledgement luokan avulla tiedetään kaikissa tapauksissa kuinka viestinkäsittely toteutui.

## Viestien kehyskerrokset

### Siirtokehys – normaalit sanomat



Kuva 6 – siirtokehyksen rakenne

Siirtokehyksen englanninkielinen termi on transmission wrapper. Siirtokehyksiä on periaatteessa kahta päätyyppiä, normaali siirtokehys ja batch-kehys. Batch-kehys sisältää useita siirtokehyksiä ja batch-lähetystä käytetään sanomien niputtamiseen

Normaalisanoman siirtokehyksen perusrakenne selviää kuvasta 6. Aina pakolliset elementit erottaa elementtilaatikon yhtenäisestä reunaviivasta. R-MIM on MCCI\_RM000100 ja se on liitteenä tiedostossa MCCI\_RM000100.png. Itse asiassa kunkin interaktion kohdalla määritellään käytettävä siirtokehys, mutta edellä mainittu on perussanomien pohjamääritys. Interaktiokohtaisessa siirtokehyksessä jokin pohjamäärityksessä vapaaehtoisena oleva tieto saattaakin olla pakollinen.

Sovellustason vastaus/kuittaussanomilla on oma siirtokehyksensä MCCI\_RM000300 ja vastaanottokuittauksilla MCCI\_RM000200. Normaalissa siirtokehyksissä (ei batch) on siis 3 erilaista alityyppiä.

Siirtokehyksen tärkeimmät tiedot ovat tiedot lähettäjästä, vastaanottajasta ja interaktiosta. Kaikkiin HL7-interaktioihin kuuluu siirtokehys.

Seuraavissa kappaleissa on selitetty tärkeimpien tietokenttien käyttö.

**infrastructureRootElements:**

Alla lueteltujen kenttien lisäksi kuvassa 6 ja muissa rakennekuvissa on infrastructureRootElements kohdan alla kolme elementtiä: realmCode, typeId ja templateId. InfrastructureRoot luokka RIM:ssä on abstrakti luokka, josta kaikki muut luokat periytyvät.

IntrastruktureRoot luokalla on kolme attribuuttia, jotka ovat käytettävissä kaikissa luokissa, myös siirtokehyksen luokissa: realmCode, typeId ja templateId ovat vapaaehtoisia elementtejä XML sanomissa, lisäksi infrastructureRoot luokasta periytyy nullFlavor attribuutti, jonka avulla kaikille elementeille saadaan hl7:n null arvot käyttöön.

InfrastructureRootElements elementit eivät ole yleensä mukana sanomissa, profileId korvaa typeId:n ja templateId:n ja realmCode näkyy interaktion ja muiden määritysten FI päätteestä.

* realmCode (CS): realm eli käyttöalue, Suomen koodi on "FI", universaali on "UV".
* typeId (II): standardin mukaiselle sanomarakenteelle voidaan määritellä lisärajoituksia ja typeId kentässä välitetään rajoitetulle rakenteelle annettu OID tunnus. Kuten profileId ks 3.1.6
* templateId (II): template määrittely on yksi keino määrittää rajoituksia sanomarakenteelle ja käytettäville koodistoille ym.

#### id (pakollinen, II)

Id on sanoman varsinainen tunnistenumero. Sen tietotyyppi on II ja 2.x vastinkenttä on MSH-10. Jos suinkin mahdollista, id:llä tulisi olla sama arvo kuin SOAP-headerin MessageId-kentällä. (Katso 3.2.4 osoitteistusprofiili). Id on OID-koodi, jossa tunnistussolmu on 18 ja id on järkevää generoida lähettävän järjestelmän osapuolitunnisteen alle.

HL7:n II tietotyyppi sallii myös UUID (Universal Unique Identifier) muotoisen arvon. Id kentän ei tarvitse olla OID, vaan se voi olla myös UUID, koska ei ole kyse pitkäaikaissäilytyksestä. UUID:tä voidaan käyttää, jos yksilöllisen OID:in luonti on vaikeaa. Huom. UUID:n käyttö ei ole sallittua Kanta-palveluihin lähetettävissä HL7 V3 sanomissa, ainoastaan OID muoto on sallittu.

Esimerkki:

<id root=”1.2.246.777.10.6280613.18.2004.225.2004.21221” />

ID:n pitää olla uniikki eikä samaa id:tä käytetä koskaan uudestaan toiselle sanomainstanssille. OID:in käyttö takaa näiden perusvaatimuksien noudattamisen. Täten id:n avulla voidaan havaita duplikaatit. Id:n generointi sanomaliikenteen osapuolistunnisteen alle esim. vuosisarjana auttaa myös sanomien pitämisessä järjestyksessä. Vuosisarja ei häiriinny kellopohjaisessa generoinnissa myöskään kesä/talviaikavaihdoksissa.

Id on yksi sanomanvälityksen hallinnan keskeisimpiä tietoja, sillä vastaanottokuittaukset ja sovellustason kuittaukset kohdistetaan alkuperäiseen sanomaan juuri tämän id:n avulla.

Id kenttä voi olla:

1. Yksikäsitteinen numeerinen OID. OID:ia käytettäessä voidaan käyttää root:ia ja extensionia, suositus tässä kohdin kuitenkin on, että käytetään pelkästään root:ia. Tärkeintä on, että root ja extension tulkitaan aina yhdessä. HUOM. Tämä vaihtoehto 1 on ainoa Kanta-palveluiden hyväksymä muoto HL7 V3 sanomissa!
2. Root on OID ja extension sisältää UUID:n
3. Koko id on UUID

UUID/GUID tyyppistä tunnistetta voidaan käyttää tässä id:ssä, koska kyseessä on kertakäyttöinen tunniste. Huom. UUID:n käyttö ei ole sallittua Kanta-palveluihin lähetettävissä HL7 V3 sanomissa, ainoastaan OID muoto on sallittu. Pysyvissä ja säilytettävissä asiayhteyksissä (kuten esim. asiakirjan tunniste) käytetään JHS 159 suosituksen mukaisesti muodostettua OID-tunnusta.

#### creationTime (pakollinen TS)

CreationTime on sanoman luontiaika. Tietotyyppi on TS ja 2.x vastine MSH-7. Käytämme sekunnin tarkkuutta.

Esimerkki:

<creationTime value=”20051211122853”/>

#### securityText (vapaaehtoinen ST)

Tietotyyppi on ST. Ei ohjeistusta toistaiseksi. Käytetään tietoturvan implementoinnissa. 2.x vastine MSH-8.

#### versionCode (vapaaehtoinen CS)

HL7 version numero. 2.x vastine MSH-12. Käytettävä versio pitää ohjeistaa sovellusaluekohtaisissa implementointioppaassa. Koodisto on HL7StandardVersionCode, arvot ovat toistaiseksi avoinna mutta ilmeisesti muotoa ”V3-2006” . Kansallisessa terveyshankkeessa käytetään juuri arvoa ”V3-2006”, koska sen sanomamääritykset perustuvat vuoden 2006 Normative Editioniin.

#### interactionId (pakollinen II)

InteractionId on sanomaan liittyvän interaktion tunnus, esim. REPC\_IN004110. Vastine 2.x:ssä on MSH-9. InteractionId on headerin tärkeimpiä tietoja, sillä se määrittelee mistä koko sanomassa on kyse ja millä tavalla vastaanottavan sovelluksen tulee toimia sanoman vastaanotettuaan.

Esimerkki:

<interactionId root=”2.16.840.1.113883.1.6” extension=”REPC\_IN004110”/>

Interaktioiden OID-koodi on 2.16.840.1.113883.1.6 kv. HL7:n määrittelemille interaktioille sekä myös Suomessa määritellyille interaktioille.

#### profileId (vapaaehtoinen SET(II) )

ProfileId ilmoittaa mihin implementointioppaisiin sanomanlähetys perustuu. Tietotyyppi on II ja elementti on toistuva. Arvona on tämän dokumentin OID ja sovellettavien implementointioppaiden OID:dit.

<profileId root="1.2.246.777.11.2008.10"/>

#### processingCode (pakollinen CS)

Tällä elementillä määritellään käyttöympäristö. Vastine 2.x:ssä on MSH-11, komponentti 1.

Arvot ovat (samat kuin 2.x:ssä):

P (production) - tuotanto

D (debugging) - testi

T (Training) - koulutus

Esimerkki:

<processingCode code=”P”/>

#### processingModeCode (pakollinen CS)

Tällä elementillä määritellään prosessointitapa. Vastine 2.x:ssä on MSH-11, komponentti 2.

Sallitut arvot ovat:

A (archive) - arkistointi

I (initial load) - peruslataus (esim. massalataus arkistosta)

R (restore from archive) - palautus arkistosta

T (current processing) - normaali prosessointi

Yleensä käytämme arvoa T.

Esimerkki:

<processingModeCode code=”T”/>

Luokituksen arvot ovat samat kuin HL7 versiossa 2.3.

Peruslataus tarkoittaa järjestelmän käyttöönottovaiheessa tapahtuvaa sanomaliikennettä, jolla järjestelmän tietosisältö ladataan tuotantokäytön edellyttämälle tasolle. Yleensä tässäkin tapauksessa on käytetty normaalin prosessoinnin arvoa T. Arkistointi tarkoittaa sitä, että sanomat viedään säilytykseen määrätyksi ajaksi auditointia varten. Tätä ei ole tiettävästi käytetty Suomessa, sillä toimenpide suoritetaan tietokantojen avulla.

#### acceptAckCode (pakollinen, CS)

Tämän elementin avulla voidaan pyytää sovellusadapterilta vastaanottokuittausta (Accept Ack kuittausta). Vastine 2.x:ssä on MSH-15 ja arvotkin ovat samat.

Arvot ovat:

AL (always) - aina

ER (error/reject only) - vain virhe tai hylkäystilanteessa

NE( never) - ei koskaan

Esimerkki:

<acceptAckCode code=”ER”/>

Vastaanottokuittaussanomassa (siirtokehys MCCI\_RM000200, interaktio MCCI\_IN000002UV01) voidaan käyttää pelkästään arvoa NE, koska sille ei voi pyytää vastaanottokuittausta. Näin estetään loputtoman ketjun syntyminen.

(Sovellustason interaktiossa voidaan kuitenkin määritellä, että sovellustason kuittaussanoma kuitataan toisella sovellustason kuittaussanomalla).

Esim. synkronisessa web services-liikenteessä ainoa järkevä arvo on ”ER”. Synkronista web services-liikennettä on selitetty kohdassa 3.2.3.3.

#### sequenceNumber (vapaaehtoinen INT)

2.x vastine on MSH-13. Käytössä, jos käytetään sequence number protokollaa. Tämä protokolla takaa, että vastaanottava sovellus saa sanomat oikeassa järjestyksessä.

Kuten aiemmin on jo mainittu tämän kentän käyttöä ei suositella.

#### attachmentText (vapaaehtoinen)

AttachmentText (toistuva) sisältää sanomaan kuuluvat liitetiedostot, joihin voidaan viitata sanoman muista osista ED tietotyypin reference mekanismin avulla, tai tämä elementti voi sisältää viittauksen liitetiedostoon url:nä reference-elementissä. (Uusimmissa ballot versioissa attachmentText on oma luokka ja sillä on id attribuutti.)

#### receiver

Tällä elementillä kuvataan vastaanottaja. Vastine 2.x:ssä on MSH-5 ja MSH-6.



Kuva 7 – vastaanottajan tietojen rakenne

Receiverin typeCode=”RCV”. Entityn device elementeistä käytetään pelkästään id:tä, jonka tietotyppi on II. Id on vastaanottajan OID-koodi. Vastaanottajan tunnistamisessa käytettävän OID-koodin tunnistussolmu on 18. (Katso myös 3.2.4 osoitteistusprofiili). Vastaanottajan käyttämän OID-koodin määrää aina vastaanottaja.

Esimerkki:

<receiver typecode=”RCV”>

<device>

<id root=”1.2.246.777.10.6280613.18.2004.225”/>

</device>

</receiver>

Vastaanottajan tunniste on lopullisen vastaanottajan tunniste, vaikka välissä tiedettäisiinkin olevan reitittäviä sovelluksia.

**Receiver/device/id:n käyttö Kanta-palvelujen kanssa kommunikoitaessa**

Kanta-palvelujen kanssa käytävän sanomaliikenteen osalta receiver/device/id:n käytöstä on olemassa oma erillinen ohjeistusdokumentti (Osapuolitiedot Kanta-sanomaliikenteessä), joka on saatavissa www.kanta.fi –portaalin arkkitehtuuriosiosta.

#### respondTo

RespondTo tarkoittaa vastausosoitetta, johon alkuperäisen interaktion vastaus sovellustasolta lähetetään. Jos tätä tietoa ei ole olemassa, käytetään sender-tietoa vastauksen lähettämiseen. RespondTo:n avulla alkuperäinen lähettäjä voi pyytää vastaanottajaa lähettämään vastaussanomat kolmannelle osapuolelle.

RespondTo:n typeCode=”RSP”. Entityn device elementeistä käytetään pelkästään id:tä, jonka tietotyppi on II. Id on vastaanottajan OID-koodi. Vastaanottajan tunnistamisessa käytettävän OID-koodin tunnistussolmu on 18.

#### sender

Lähettäjä kuvataan vastaavalla tavalla kuin vastaanottaja, OID-koodilla devicen id-elementissä. Senderin typeCode=”SND”. Versio 2.x vastine on MSH-3 ja MSH-4. Myös yhteystietojen ilmoittaminen on mahdollista, esim. sovellusliitännästä vastaavan tahon yhteystiedot mahdollista virheselvittelyä varten.



Kuva 8 - lähettäjä

**Sender/device/id:n käyttö Kanta-palvelujen kanssa kommunikoitaessa**

Kanta-palvelujen kanssa käytävän sanomaliikenteen osalta sender/device/id:n käytöstä on olemassa oma erillinen ohjeistusdokumentti (Osapuolitiedot Kanta-sanomaliikenteessä), joka on saatavissa www.kanta.fi –portaalin arkkitehtuuriosiosta.

#### attentionLine (vapaaehtoinen)

AttentionLine sisältää informaatiota muodossa tiedon nimi, tiedon arvo. Tähän elementtiin voidaan laittaa esim. reitityksessä tarvittavia tietoja, jos varsinainen payload on salattu. AttentionLine ei vaikuta varsinaisen sanoman prosessointiin. Jos tätä elementtiä jatkossa aiotaan hyödyntää, niin sen käytöstä tullaan antamaan tarkempia ohjeita.

#### ControlActProcess

Tämä elementti kuvaa tapahtumaan (triggeriin) liittyvät yleistiedot. Varsinainen sanoma on ripustettu tämän elementin alle. Kyseessä on siis aiemmin mainittu kontrollikehys, joka on kuvattu luvussa 2.3.3.

### Siirtokehys – kuittaussanomat

Käytettävä siirtokehys pitää aina tarkistaa käytettävästä interaktiosta. Sovellustason kuittauksien siirtokehys MCCI\_RM000300 on hyvin samanlainen kuin perussanomien siirtokehys MCCI\_RM000100. Lisänä on acknowledgement-luokka. Sama luokka on myös sovellusadapterien vastaanottokuittauksien (Accept Ack) siirtokehyksessä MCCI\_RM000200. Vastaanottokuittauksien siirtokehyksestä sen sijaan on jätetty pois monia tarpeettomia elementtejä, myös kontrollikehys, koska varsinaista payloadia ei tarvita sovellusadapterien vastaanottokuittauksissa. Vastaanottokuittausta on käsitelty kohdissa 2.6 ja 3.1.2.2.



Kuva 9 – vastaanottokuittauksen siirtokehys



Kuva 10 - kuittaustiedot

#### acknowledgement – typeCode (pakollinen, CS)

Elementissä acknowledgement kerrotaan kuittauksen varsinainen arvo elementissä typeCode. Arvot ovat samat kuin 2.x:ssä:

Sovellustason kuittaukset:

|  |  |  |
| --- | --- | --- |
| Koodi | tarkoitus | suomennus |
| AA | Application Acknowledgement Accept | Sovellustason OK kuittaus. |
| AE | Application Acknowledgement Error | Sovellustason virhe. (Sanomaa ei kannata lähettää uudestaan). |
| AR | Application Acknowledgement Reject | Sanomakäsittely ei onnistunut. (Lähettäjä tekee parametroidun määrän uudelleenlähetyksiä) |

Vastaanottokuittaukset:

|  |  |  |
| --- | --- | --- |
| Koodi | tarkoitus | suomennus |
| CA | Accept Acknowledgement Commit Accept | Sovellusadapteri on vastaanottanut sanoman onnistuneesti. |
| CE | Accept Acknowledgement Commit Error | Sovellusadapteri ei pystynyt käsittelemään sanomaa, uudelleenlähetystä kannattaa yrittää. |
| CR | Accept Acknowledgement Commit Reject | Kyseisen tyyppisen sanoman käsittely ei onnistu tässä sovelluksessa. Uudelleenlähetyksiä ei kannata yrittää. |

#### expectedSequenceNumber (vapaaehtoinen, INT)

Odotettu järjestysnumero, jota käytetään sequence-number protokollassa. Versio 2.x vastine on MSA-4.

#### messageWaitingNumber (vapaaehtoinen, INT)

Tämä elementti kertoo, paljonko kuittaavalla sovelluksella on sanomia jonossa. Tätä tietokenttää käytetään vain silloin, kun sanomat noudetaan pollaamalla lähettäjän jonosta.

#### messageWaitingPriorityCode (vapaaehtoinen, INT)

Tämä elementti ilmoittaa, mikä on kuittaavan sovelluksen sanomajonon sanomien korkein prioriteetti. Tätä tietokenttää käytetään vain silloin, kun sanomat noudetaan pollaamalla lähettäjän jonosta.

#### targetMessage – id (pakollinen, II)

Kuitattavan sanoman tunnistenumero ilmoitetaan elementissä <targetMessage><id>. Vastine 2.x:ssä on MSA-1.

Esimerkki:

<acknowledgement >

<typeCode code=”AA”/>

<targetMessage>

<id root=”1.2.246.777.10.6280613.18.2004.225.2004.21221” />

</targetMessage>

</acknowledgement>

#### acknowledgementDetail (vapaaehtoinen)

Tämän toistuvan elementin avulla kerrotaan kuitattavasta sanomasta, mikä virhe oli kyseessä ja missä kohdassa. Tämä elementti vastaa 2.x:n tietoryhmää ERR. Elementin kaikki osat ovat myös vapaaehtoisia.

|  |  |  |  |
| --- | --- | --- | --- |
| elementti | 2.x  vastine | tietotyyppi | käyttötarkoitus/arvot |
| **TypeCode** | ERR-4 | CS | E=error, I=Information, W=Warning  Koodi kertoo virhetilanteen vakavuuden |
| **Code** | ERR-3+  ERR-5+  ERR-6 | CE | Virheilmoitus koodattuna(mielellään) tai pelkkänä tekstinä. Koodisto on AcknowledgementDetailCode (CWE). |
| **text** | ERR-7 | ED | Virheeseen liittyvä lisätieto |
| **Location** | ERR-2 | SET(ST) | Virheen sijainti |

Tässä elementissä ilmoitetaan tekniset virheet sekä vastaanottokuittauksessa että sovellustason kuittauksessa. Sovellustason virheet ilmoitetaan kontrollikehyksessä.

### Kontrollikehys - normaalisanomat

**

kuva 11 - kontrollikehys

Teknisesti kontrollikehys (control act wrapper) on siirtokehyksen (transmission wrapper) yksi elementti. Sen pääluokka on ControlActProcess. Varsinainen sanoma sijoitetaan kontrollikehyksen alle (act-relationship **subject ).** Kontrollikehyksessä kerrotaan triggeriin liittyvät tiedot. Vastaavuus 2.3:ssa on EVN. D-MIM on MCAI\_DM700200 ja se on liitetiedossa MCAI\_DM700200.png. Kunkin sovellusalueen komiteat määrittelevät kontrollikehyksen soveltamisen. Tämän vuoksi yleisessä pohjamäärityksessä suurin osa kentistä on jätetty vapaaehtoiseksi. Control act on kaikille perussanomille samanlainen.

#### classCode (pakollinen, CS)

classCode=”CACT” (vakio)

#### moodCode (pakollinen, CS)

moodCode=”*EVN*” (muuttuva)

MoodCode riippuu kyseessä olevasta interaktiosta, mutta yleensä arvo on EVN.

Kyselysanomissa käytetään arvoa RQO.

#### Id (SET(II) )

Tässä elementissä ilmoitetaan tapahtuman yksikäsitteinen tunnus sovellustasolla (OID-koodilla). Elementti on toistuva.

#### code (CD)

Elementtiin code sijoitetaan tieto triggeristä, esim. REPC\_TE004110 .

Esimerkki:

<ControlActProcess classCode=”CACT” moodCode=”EVN”>

<code code=”REPC\_TE004110“ codeSystem=“2.16.840.1.113883.1.18“/>

</ControlActProcess>

Kyseessä on triggeri, joka aiheutti juuri tämän sanoman lähetyksen.

Triggereiden OID-koodi on 2.16.840.1.113883.1.18 kv. HL7:n määrittelemille triggereille sekä myös Suomessa määritellyille triggereille.

#### text (ED)

Tekstimuotoinen kuvaus tapahtumasta.

#### effectiveTime ( IVL(TS) )

Tapahtuman aika. Se on yleensä eri aika kuin sanoman lähetysaika siirtokehyksessä.

#### priorityCode ( SET(CE) )

Prioriteettikoodien vaihtoehdot löytyvät sanastosta ActPriority. Normaaliarvo on R (routine).

#### reasonCode ( SET(CE) )

Syykoodit löytyvät sanastosta ActReason. Esimerkiksi FIN tarkoittaa, että potilaalla ei ole vakuutusta eikä maksukykyä. Jos käytetään elementtiä DetectedIssueEvent (reasonOf alla), on tämä kohta jätettävä tyhjäksi.

#### languageCode ( CE)

Kielikoodi koodistosta IETF RFC 1766.

#### Osapuolet

Kontrollikehyksessä voidaan ilmoittaa seuraavat (vapaaehtoiset) osapuolet, joiden käyttö määritellään tapauskohtaisesti. Yleensä käytetään organisaatiotason tietoa. Organisaation tiedot tulevat R\_AssignedPerson CMET:in Organization luokkaan, kyseisen CMET:in henkilön tiedot jätetään tällöin usein pois.

|  |  |
| --- | --- |
| authorOrPerformer | Suorittaja tai tiedosta vastaava |
| overseer | Valvoja |
| dataEnterer | Tallentaja |
| informationRecipient | Tiedon vastaanottaja |
|  |  |

#### subject

Subject act-relationship:in alle sijoitetaan varsinainen sanoma eli payload. Sanomia saa olla useita, mutta niiden pitäisi olla samaa tyyppiä. Attribuutti contextConductionIndicator on aina **false**, koska on tarkoitus, että kontrollikehyksen tiedot eivät periydy varsinaiseen payloadiin.

#### reasonOf

Teknisluontoiset kuittaustiedot sijoitetaan siirtokehykseen. Jos kyseessä on prosessiin liittyvä virhetilanne (esim. pyyntöä ei voi enää perua, koska näyte on jo otettu), niin tiedot sijoitetaan kontrollikehyksen elementtiin reasonOf. TypeCode on vakio ”RSON” ja contextConductionIndicator on arvossa false. ReasonOf sisältää toistuvan elementin detectedIssueEvent, jolla varsinaiset virhetiedot esitetään.

Classcode on vakio ”ALRT” ja moodCode on vakio ”EVN”. DetectedIssueEvent sisältää seuraavat tiedot:

id (II) - virheen yksikäsitteinen tunniste

code (CD) - virhe tyyppi koodatussa muodossa

text(ST) - virheen lisätiedot

value (ANY) - tarkka virhetieto (Observation value-tyyppisesti)

(Alkuperäiseltä lähettäjältä) vaadittavat toimenpiteet ilmoitetaan elementillä triggerFor-actOrderRequired.

Hyväksyttävät/kyseessä oleva poikkeustilanteet ilmoitetaan elementin mitigatedBy-detectedIssueManagement avulla.

Esim. negatiivisessa sovellustason kuittauksessa (detectedIssue=”lääkettä ei voida toimittaa, koska se on jo toimitettu”) voidaan esim. ilmoittaa koodattuna detectedIssueManagementissa pyytäjälle, että kaksinkertaisen toimituksen eston voi kiertää vain, jos kyseessä lääkkeiden hukkuminen tai pilaantuminen. Tällöin moodCode on DEF. Pyytäjä voi sen jälkeen lähettää uuden pyynnön, jossa detectedIssueManagementissa ilmoitetaan koodattuna, että kyseessä on lääkemääräyksen rikkomispyyntö ja syykoodi on lääkkeiden hukkuminen. MoodCode on tällöin EVN.

DetectedIssueManagementin avulla hallitaan siis poikkeamat sovituista säännöistä (hätätapauksissa). Tästä käytetään englanninkielistä termiä override request.

Näiden elementtien käyttö pitää ohjeistaa tapauskohtaisissa dokumenteissa tarkemmin. Luvussa 2.7.2 on esimerkki ajanvarausmäärittelyn käsittelytavasta.

### Kontrollikehys – keskitetyt rekisterit

Keskitettyjen rekistereiden alkuperäinen termi on Master file. Tällaisia ovat esim. keskitetty potilasrekisteri, hintarekisteri ja organisaatiorekisteri. Keskitettyjen rekistereiden sanomien kontrollikehys määritellään omassa domainissaan: MFMI.

Keskitettyjen rekistereiden tapauksessa varsinaista payloadia ei olekaan laitettu suoraan kontrollikehyksen pääluokan ControlActProcess alle, vaan välissä on luokka RegistrationProcess. Se sisältää muutoksen hallinnassa tarvittavat tiedot. Varsinainen payload voi olla roolipohjainen, jolloin se sijoitetaan stub:in RegisteredRole tilalle tai tapahtumapohjainen, jolloin varsinainen payload sijoitetaan stub:in RegisteredAct tilalle.

## Kyselysanomat

### Yleistä HL7v3:n kyselysanomista

Kyselysanomia varten HL7 versio 3:ssa on kehitetty erillinen kyselyrunko (*Query Infrastructure*). Sitä on tarkoitus käyttää kyselyiden toteutuksissa HL7v3:n eri domaineissa. Kyselyrunkoa ei voi käyttää suoraan, vaan sitä pitää laajentaa domain-kohtaisella tiedolla, eli runkoon pitää lisätä tarvittavat tietoelementit kyselyjen ja vastausten esittämiseen. Kyselyviestin elementtien pakollisuus ja parametrien väliset suhteet (esim. boolean logiikka) ilmaistaan domain kohtaisessa määrittelyssä. HL7v3 2006-standardissa kyselyrunko on määritelty omaksi domainikseen ja se löytyy kohdasta *Specification Infrastructure* ⇒ *Messaging* ⇒ *Query Infrastructure*.

Seuraavassa kuvassa 12 on esitetty yleisesti kyselyrungon määrittämät sovellusroolit, joita on kaksi kappaletta:

* QUQI\_AR000001 (QueryByParameter Placer)
* QUQI\_AR000002 (QueryByParameter Fulfiller)



Kuva 12 – kyselyrungon määrittämät sovellusroolit ja vuorovaikutukset

Kyselyrungon tukemia kyselyitä on esimerkiksi:

* Yhden tietyn potilaan kaikkia laboratoriotuloksia koskeva kysely.
* Lääkärin tai muun terveydenhuoltoalan ammattilaisen potilaslistan hakeminen tietyllä aikavälillä.
* Potilaasta riippumattoman tiedon hakeminen, esimerkiksi lääketietojen hakeminen tietylle lääkkeelle.

Kyselyrunko sisältää yleisiä perusrakenteita, joita laajentamalla voidaan esittää:

* parametri-tyyppinen kysely (*QueryByParameter*, kappale 2.4.3)
* itse kyselyn parametrit (*ParameterList*, kappale 2.4.3)
* palautettavien vastausten toivottu lajittelu (*SortControl*, kappale 2.4.3)
* kyselyn vastaukset (*QueryAck*, kappale 2.4.4)
* lisävastausten kysyminen (*Query*C*ontinuation*, kappale 2.4.5)

Kyselysanomaa rakennettaessa kyselyn määrittäjän tehtäväksi jää periaatteessa vain kyseltävän tiedon määrittäminen. Tämä voidaan tehdä listaamalla kyselyparametreja ja määrittämällä, mitä kyselyn vastauksen laatijan on palautettava näitä hakuarvoja käytettäessä kyselyn yhteydessä. HL7v3 sisältää valmiina yleisen rungon tällaisten kyselyparametrien esittämiseen (*QueryByParameter*). Tämä runko voidaan tarvittaessa määrittää kokonaan itse, jos kyselyä ei voida toteuttaa valmiilla rakenteella.

Kyselyrunko sisältää myös mahdollisuuden määrittää domain-kohtaisesti koko QueryByParameter-rakenne. Tällöin voidaan itse määritellä vielä tarkemmin halutut kyselyrakenteet, jolloin ParameterList- ja SortControl-rakenteita ei tarvitse käyttää.

Kyselysanomat ja niiden vastaukset ovat tavallisia HL7v3-viestejä (ks. kappale 2.1), eli niille määritellään aina:

* siirtokehys
* kontrollikehys (ks. kappale 2.3.3)
* itse sanoma (ks. kappaleet 2.4.3, 2.4.4, 2.4.5)

Alla olevissa luvuissa on käsitelty yksityiskohtaisemmin kyselyihin liittyvät viestit ja niissä käytettävät tietosisällöt. On huomioitava että yleisessä kyselyrungossa monet attribuutit ovat vapaaehtoisia ja tiettyyn käyttötarkoitukseen laaditussa kyselyssä voidaan attribuutteja muuttaa pakolliseksi. Myös koodattujen arvojen koodistot voivat poiketa yleisessä kyselyrungossa käytetyistä. Kyselynrunkoon määritellyt koodistot ovat kuitenkin aika kattavia, joten oletettavaa on että käyttötarkoituskohtaisissa kyselyissä käytetään pitkälti samoja koodistoja.

### Yleinen kontrollikehyksen rakenne

Kyselyrungon kontrollikehys vastaa pitkälti kappaleessa 2.3.3 esitettyä yleistä kontrollikehysrakennetta, jota on laajennettu kyselyrunkoon liittyvällä domain-kohtaisilla rakenteilla. Kyselyitä esittävissä kuvista on riisuttu pois ne yleiset rakenteet jotka on jo kuvattu aiemmissa luvuissa.

Tarkemmin sanottuna kyselyrunko sisältää kolme hieman erilaista, normaalista kontrollikehyksestä laajennettua kontrollikehystä; kyselylle, vastaukselle ja lisävastausten välittämiselle on omat kontrollikehykset. Ne on nimetty kaikki samalla tavalla (*ControlActProcess*), ja erot niiden välillä ovat kehyksen sisältämät erilaiset payload-osiot. Nämä kontrollikehykset ja niitä vastaavat payload-osiot on kuvattu erikseen kappaleissa 2.4.3, 2.4.4 ja 2.4.5.

### Query Specification-kontrollikehys

Query Specification –kontrollikehyksen avulla voidaan esittää itse kysely ja siihen liittyvät metatiedot.



Kuva 13 Kyselyrungon riisuttu RMIM QUQI\_RM020000UV01 (V3 2006 standardi)

Kuvan R-MIM kaaviossa on kyselyrakenne, jossa määritellään domain kohtaisesti pelkät parametrit. Jos koko kyselymäärittely tehdään domain kohtaisesti, käytetään R-MIMiä QUQI\_RM021000UV01. Taulukossa on selvitetty kyselysanomassa käytettävien attribuuttien merkitys.

**QueryByParameter**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribuutti | Tyyppi | Lkm | Selitys tai sallitut arvot |
| queryID | II | [0..1] | Kyselysanoman tunniste, jolla kyselyyn vastaaja voi yhdistää vastauksen alkuperäiseen kyselyyn |
| statusCode | CNE:QueryStatusCode | [1..1] | new – uusi kysely  aborted – kysely on peruutettu |
| modifyCode | CNE:ModifyIndicator | [0..1] | N – uusi kysely  M – kysely muokkaa aikaisemmin tehtyä kyselyä |
| responseElementGroupID | SET<II> | [0..\*] | Kyselyn vastauksen tyypin on oltava joku tässä luetellusta tyypeistä. |
| responseModalityCode | CNE:ResponseModality | [0..1] | B – vastaus lähetetään batch-tavalla  R – vastaus palautetaan reaaliaikaisesti  T – vastaus lähetetään sarjana erillisiä vastauksia ilman batch-tapaa. |
| responsePriorityCode | CNE:QueryPriority | [0..1] | D –vastaus voidaan lähettää myöhemmin  I – vastaus lähetetään välittömästi |
| initialQuantity | INT | [0..1] | Maksimimäärä vastauksia (lukumäärä), joita vastauksen vastaanottaja hyväksyy |
| initialQuantityCode | CWE:QueryRequestLimit | [0..1] | Mitä vastauksia (yksikköä) vastauksen vastaanottaja maksimissaan hyväksyy. Vastaussanomat voivat sisältää muutakin tietoa, kuin haluttuja vastausrakenteita. Tällä attribuutilla voidaan tunnistaa, minkä vastausyksikön perusteella vastausten lukumäärä päätellään. Koodistolle ei ole yleistä määrittelyä, joten se määritellään tapauskohtaisesti. |
| executionAndDeliveryTime | TS | [0..1] | Aika, johon mennessä vastaus tulisi palauttaa |
| parameterList | domain-kohtainen | [1..\*] | Itse kyselyparametrit, jonka perusteella kysely suoritetaan. Kts. esimerkki kappaleesta 2.4.6 |
| sortControl | SortControl | [0..\*] | Metatietoa, joka määrää kyselyn vastausten lajittelun. |
| sortControl.sequenceNumber | INT | [0..1] | SortControl-rakenteen tärkeysjärjestys, ts. missä järjestyksessä SortControl-rakenteita tulisi soveltaa vastausta koostettaessa jos rakenteita on määritelty useampi yhdelle kyselylle |
| sortControl.elementName | SC | [0..1] | RIM-elementin tunniste, jonka mukaan vastaus lajitellaan |
| sortControl.directionCode | CS | [0..1] | A – vastaus lajitellaan nousevassa järjestyksessä  D – vastaus lajitellaan laskevassa järjestyksessä  N – lajittelujärjestyksellä ei ole väliä |

### Query Response/Acknowledgement-kontrollikehys

Query Response/Acknowledgement –kontrollikehyksen avulla kyselyyn vastaaja välittää vastaukset ja siihen liittyvät metatiedot.



Kuva 14 – Query Response/Acknowledgement-kontrollikehys (riisuttu versio HL7v3 2006 –standardin vastaavasta)

Kontrollikehyksen tärkeimmät osiot ovat QueryAck ja subject. QueryAck- luokka sisältää kyselyvastaukseen liittyviä kuvailutietoja, joiden merkitys on kuvattu alla olevassa taulukossa. Subject -suhde linkittää domain kohtaisesti määriteltävän varsinaisen vastauksen kontrollikehykseen. QueryByParameter luokalla voidaan tarvittaessa ilmaista myös vastauksessa alkuperäinen kysely. **QueryAck**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribuutti | Tyyppi | Lkm | Selitys tai sallitut arvot |
| queryID | II | [0..1] | Kyselysanoman tunniste, jolla kyselyyn vastaaja voi yhdistää vastauksen alkuperäiseen kyselyyn |
| statusCode | CNE:QueryStatusCode | [0..1] | deliveredResponse – vastaus kyselyyn  aborted – vastauksen koostaminen on peruutettu  executing – kysely on parhaillaan suoritettavana |
| queryResponseCode | CNE:QueryResponse | [1..1] | AE – sovellusvirhe  NF – yhtään vastausta ei löytynyt kyselyyn  OK – yksi tai useampi vastaus löytyi kyselyyn  QE – kyselyn parametreissa oli virhe |
| resultTotalQuantity | INT | [0..1] | Kyselyyn löytyneiden vastausten kokonaislukumäärä |
| resultCurrentQuantity | INT | [0..1] | Tässä vastaussanomassa ilmoitettujen vastausten lukumäärä |
| resultRemainingQuantity | INT | [0..1] | Vielä palauttamattomien vastausten lukumäärä |

**Subject**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribuutti | Tyyppi | Lkm | Selitys tai sallitut arvot |
| typeCode | CNE:SUBJ | [1..1] | ”SUBJ” |
| contextConductionInd | BL | [1..1] | ”false” |
| act | domain-kohtainen | [1..1] | Itse vastaukset. Tämä rakenne voi olla myös toisen niminen. Kts. esimerkki kappaleessa 2.4.7. |

### Query Continuation/Cancel-kontrollikehys

Query Continuation/Cancel –kontrollikehyksen avulla kyselyn alkuperäinen tekijä voi joko ilmoittaa haluavansa vielä palauttamattomia lisävastauksia tai lopettaa aloitetun kyselyn, jolloin palauttamattomat lisävastaukset hylätään.



Kuva 15 – Query Continuation/Cancel-kontrollikehys (riisuttu versio HL7v3 2006 –standardin vastaavasta)

**QueryContinuation**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribuutti | Tyyppi | Lkm | Selitys tai sallitut arvot |
| queryID | II | [0..1] | Kyselysanoman tunniste, jolla kyselyyn vastaaja voi yhdistää vastauksen alkuperäiseen kyselyyn |
| statusCode | CNE:QueryStatusCode | [1..1] | aborted – kysely on peruutettu  waitContinuedQueryResponse – lisävastauksia odotetaan |
| startResultNumber | INT | [0..1] | Lisävastausten koostamisessa mahdollisesti tarvittava metatieto, mistä lähtien vastauksia palautetaan |
| continuationQuantity | INT | [0..1] | Lisävastausten koostamisessa mahdollisesti tarvittava metatieto, montako vastausta palautetaan seuraavaksi.  Jos arvona on 0, niin pyydetään kyselyn keskeyttämistä. |

### Esimerkki kyselyviestin ParameterList-rakenteesta

Tässä kappaleessa on esitetty esimerkkinä Patient Administrationin Person-aihepiiristä löytyvän FindCandidates-kutsun käyttämästä tietorakenteesta, jolla ilmaistaan potilashakua. Haettavat potilaat ilmaistaan kyselyrungon mukaisella ParameterList-rakenteella, joka siirretään asianmukaisen QueryByParameter-payloadin ja QuerySpecification-kontrollikehyksen sisällä.



Kuva 16 – FindCandidates haussa käytettävä parametrilista

### Esimerkki kyselyvastauksen Subject-rakenteesta

Tässä kappaleessa on esitetty esimerkkinä osa Patient Administrationin Person-aihepiiristä löytyvän FindCandidates-kutsun vastauksen käyttämästä tietorakenteesta, jolla ilmaistaan potilaslistaa. Potilaslista ilmaistaan Subject-rakenteella, joka siirretään asianmukaisen Query Response-kontrollikehyksen sisällä payload-osiona. Varsinainen vastaus on registrationEvent, joka on sijoitettuna kontrollikehyksen subject rakenteen alle. RegistrationEventin sisällä varsinaiset kyselyparametrit täyttäneet ehdokkaat löytyvät subject1 elementistä.



Kuva 17 – FindCandidates haun vastausrakenne



Kuva 18 – Subject1 rakenteen yksityiskohtainen sisältö

Kuvassa 18 on tunnistetun henkilön esittämiseen käytetty XML-rakenne. Kuvassa alimalla tasolla oleva identifiedPerson-elementti esitetään viestityypin QUPA\_MT101104.Person avulla, jonka XML-rakenne on alla olevassa listauksessa.

<xs:complexType name="QUPA\_MT101104.Person">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="name" type="PN" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="desc" type="ED" minOccurs="0"/>

<xs:element name="telecom" type="TEL" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="administrativeGenderCode" type="CE" minOccurs="0"/>

<xs:element name="birthTime" type="TS" minOccurs="0"/>

<xs:element name="deceasedInd" type="BL" minOccurs="0"/>

<xs:element name="deceasedTime" type="TS" minOccurs="0"/>

<xs:element name="multipleBirthInd" type="BL" minOccurs="0"/>

<xs:element name="multipleBirthOrderNumber" type="INT" minOccurs="0"/>

<xs:element name="addr" type="AD" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="maritalStatusCode" type="CE" minOccurs="0"/>

<xs:element name="asOtherIDs" type="QUPA\_MT101104.OtherIDs" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asEmployee" type="QUPA\_MT101104.Employee" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asCitizen" type="QUPA\_MT101104.Citizen" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asStudent" type="QUPA\_MT101104.Student" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asMember" type="QUPA\_MT101104.Member" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="birthPlace" type="QUPA\_MT101104.BirthPlace" nillable="true" minOccurs="0"/>

<xs:element name="languageCommunication" type="QUPA\_MT101104.LanguageCommunication" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="classCode" type="EntityClass" use="optional" fixed="PSN"/>

<xs:attribute name="determinerCode" type="EntityDeterminer" use="optional" fixed="INSTANCE"/>

</xs:complexType>

## Batch-sanomat

Batch-sanomia ei ole määritelty normatiivisessa V3 2006-standardissa mutta May 2007 ballot-paketin Transmission Infrastructure-osiossa on Batch Wrapper topic, josta tässä esitettävä aineisto on peräisin.

RIM sisältää Batch-luokan kuten myös siirtokehyksen sovellusalueen DMIM, joka on kuvassa MCCI\_DM000000.png.

V3:ssa Batch-sanomien rakenne ja käsittely on varsin yksinkertainen: batch-interaktio sisältää muita interaktioita, jotka käsitellään kuin ne olisi lähetetty erillisinä.

Esim. Batch-interaktio Send Batch (MCCI\_IN200100UV), joka sisältää RCMR\_IN000002UV01 interaktioita. Batch rakenne muodostaa tavallaan kolmannen wrapper-kerroksen (siirtokehyksen) sanomien ympärille.

<MCCI\_IN200100UV>

<id>

<creationtime>

batch luokan attribuutit

ja siihen liittyvät muut kehyksen luokat

<receiver>

<sender>

<!--tähän interaktiot -->>

<RCMR\_IN000002UV01>

…………..

<RCMR\_IN000002UV01/>

<RCMR\_IN000002UV01>

…………..

<RCMR\_IN000002UV01/>

<MCCI\_IN200100UV>

Yhdessä batch-sanomassa voidaan lähettää erityyppisiä interaktioita, batch-sanoma voi olla myös tyhjä.

Batch-sanomaliikenteen toteuttaminen vaatii enemmän ohjeistusta implementointioppaassa kuin normaali V3-sanomaliikenne. Batch:in käyttö vaatii sovellusaluekohtaisen rajauksen. V3:ssa käytettävät kuittausmekanismit on kehitetty yksittäisten sanomien kuittaukseen ja batch-liikenne tuo lisää tilanteita, joihin standardi ei anna ohjeita. Samoin batch-interaktioihin liittyvät vastaussanomat voidaan lähettää yksittäisinä sanomia tai ne voidaan lähettää vastausbatchinä (Response Batch).

Ballot-paketissa on määritelty kaksi batch sanomakuvausta: **Batch RMIM** **(MCCI\_RM200100UV)** ja **Response Batch** **(MCCI\_RM200101UV),** jotka sisältävät Batch-luokan lisäksi samoja luokkia kuin normaalisanomien ja kuittaussanomien siirtokehykset.

Batch RMIM on tiedostossa MCCI\_RM200100.png ja Response Batch on tiedostossa MCCI\_RM200101.png

### Batch siirtokehys



kuva 19 Batch-siirtokehys

Suurin osa kentistä on samoja kuin siirtokehyksessä, näistä ei ole selitystä.

#### Id (pakollinen II)

#### creationTime (pakollinen TS)

#### securityText (vapaaehtoinen ST)

#### responseModeCode (pakollinen CS)

Tietotyyppi on CS. Tämä tieto kertoo vastaanottajalle, kuinka batch-interaktio ja sen sisältämät sanomat tulee käsitellä.

Arvot ovat seuraavat:

I (immediate) - vastaanottava sovellus vastaa synkronisesti

D (deferred) - vastaanottava sovellus vastaa sitten kun ehtii (default)

Q (queue) - vastaanottava sovellus laittaa vastauksen jonoon ja toinen osapuoli käy hakemassa vastauksen jonosta (ei käytössä Suomessa)

Esim. <responseModeCode code="D"/>

Saapuviin batch-interaktion sisältämiin interaktioihin lähetetään sovellustason vastaussanoma tai ei mitään, riippuen siitä, mitä interaktiolle on standardissa määritelty. Lähetetäänkö vastaussanomat erillisinä interaktioina vai kootaanko ne Response Batch- sanomaksi on sovittava ko. integraation implementointioppaassa.

#### versionCode (vapaaehtoinen CS)

#### interactionId (pakollinen II)

May2007 ballot paketti määrittelee kolme interaktiota, **Send Batch** (MCCI\_IN200100UV), **Send Response Batch** (MCCI\_IN200101UV) ja **Send message Adapter Batch Acknowledgement** (MCCI\_IN200102UV).

#### referenceControlId (required, voi olla null, II)

Ilmoittaa batchin ensimmäisen lähetyksen Id:n, ei käytetä lähetettäessä batchia ensimmäistä kertaa.

#### name (vapaaehtoinen SC)

Batchia käsittelevä sovellus voi käyttää hyväkseen tätä kenttää batchin tyypin määrittämiseen.

Tietotyyppi on SC (Character String with Code), joka on string-tyyppi (ST), johon on liitetty vapaaehtoinen koodi. Tekstin on oltava aina mukana, koodi on vapaaehtoinen.

Koodisto on usein paikallinen eli se määritellään implementointioppaassa.

Esim. <name code="koodi" codeSystem="1.2.3.x.x" codeSystemName="Oma Batch koodisto">Nimi</name>

#### batchComment (vapaaehtoinen, toistuva SET<ST>)

Voidaan käyttää batchiin liittyvien kommenttien esittämiseen, toistuva.

#### transmissionQuantity (required INT)

Ilmoittaa batchin sisältämien interaktioiden lukumäärän.

#### batchTotalNumber (vapaaehtoinen, toistuva SET<INT>)

Paikallisesti voidaan sopia miten batchin sisältämistä interaktioista lasketaan yksi tai useampia kontrollilukuja.

#### Message

Toistuva message elementti kuvaa batchin sisältämiä sanomia, jotka ovat erillisiä interaktioita. Kaikki V3-sanomat kuuluvat RIM:n Message luokkaan.

Todellisessa batch-sanomassa ei ole <message>-elementtejä vaan schemaa on muutettava, jotta interaktiot voidaan sisällyttää batch-wrapperiin. Yksi mahdollisuus on lisätä schemaan <any>-elementti, jonka paikalle batchin sisältämät interaktiot tulevat. Toisin kuin alkuperäinen message-elementti, tämä any-elementti kannattaa sijoittaa batch-kehyksen loppuun, jotta esim. sender- ja receiver-elementit pystytään validoimaan XML-parserissa.

Esim. muutettu schema

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II"/>

<xs:element name="creationTime" type="TS"/>

<xs:element name="securityText" type="ST" minOccurs="0"/>

<xs:element name="responseModeCode" type="CS"/>

<xs:element name="versionCode" type="CS" minOccurs="0"/>

<xs:element name="interactionId" type="II"/>

<xs:element name="referenceControlId" type="II" minOccurs="0"/>

<xs:element name="name" type="SC" minOccurs="0"/>

<xs:element name="batchComment" type="ST" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="transmissionQuantity" type="INT" minOccurs="0"/>

<xs:element name="batchTotalNumber" type="INT" minOccurs="0" maxOccurs="unbounded"/>

<!-- message elementtiä ei esiinny todellisessa batch sanomassa

<xs:element name="message" ... minOccurs="0" maxOccurs="unbounded"/>

-->

<xs:element name="receiver" ... />

<xs:element name="respondTo" ... />

<xs:element name="sender" type="MCCI\_MT200100UV.Sender"/>

<xs:element name="attentionLine" ... />

<!-- any elementti lisätty, tähän tulevat batchissa lähetettävät interaktiot -->

<xs:any />

<!-- Message Interactions HERE

-->

</xs:sequence>

#### receiver (pakollinen)

#### respondTo (vapaaehtoinen)

#### sender (pakollinen)

#### attentionLine (vapaaehtoinen)

### Response Batch siirtokehys

Response Batch-siirtokehys MCCI\_RM200101UV sisältää samat elementit kuin batch- siirtokehys MCCI\_RM200100UV sekä lisäksi pakollisen acknowledgement-elementin (kuten yleensä kuittaussanomien siirtokehykset)..

Response-batchissä palautetaan kuitattavan batchin sisältämien interaktioiden sovellustason vastausinteraktiot.

Response batchia vastaava kuitattava batch sanoma ilmoitetaan pakollisen **acknowledgement/targetBatch/id** -elementin avulla.



kuva 20 Batch acknowledgement - elementti

#### message

R-MIM:in message elementtiä ei esiinny todellisessa sanomassa. Sen korvaa XML-sanomassa batch-kehyksen lopussa oleva any-elementti, joka sisältää siirrettävät interaktiot.

#### acknowledgement – typeCode (pakollinen, CS)

Käytännössä on aina “AA”, batchin sisältämillä interaktioilla on omat kuittauskoodinsa.

#### acknowledgement – targetBatch – id (pakollinen, II)

Id elementissä ilmoitetaan kuitattava batchi.

### Batch Interaktiot

Ballot-paketti määrittelee kolme interaktiota, joissa sanomakuvauksia käytetään: **Send Batch** (MCCI\_IN200100UV), **Send Response Batch** (MCCI\_IN200101UV) ja **Send message Adapter Batch Acknowledgement** (MCCI\_IN200102UV).

#### Send Batch (MCCI\_IN200100UV)

Käytetään batch sanoman lähettämiseen. Siirtokehys on MCCI\_MT200100UV.

Koska siirtokehys ei sisällä acceptAckCode attribuuttia (eli halutaanko batchille vastaanottokuittaus) on implementointioppaassa sovittava käytetäänkö koko batch-sanomalle Accept Ack kuittausta (MCCI\_IN000002UV01).

#### Send Response Batch (MCCI\_IN200101UV)

Käytetään batch vastaussanoman lähettämiseen ja sisältää alkuperäisen batch sanoman sisältäneiden interaktioiden sovellustason vastaussanomat.

Siirtokehys on MCCI\_MT200101UV

Lähetetäänkö vastaussanomat erillisinä interaktioina vai tällä interaktiolla on sovittava implementointioppaassa.

Yhteen batch lähetykseen voidaan vastata vain yhdellä Response Batch interaktiolla eli kaikki vastausinteraktiot on koottava yhteen pakettiin jos tätä interaktiota käytetään vastauksessa.

Query sanomat voivat myös pyytää vastausta batchinä.

Mikään ei kuitenkaan estä käyttämästä send batch-wrapperiä (MCCI\_IN200100UV ) myös sovellustason vastaussanomien paketoimiseen, etenkin jos koko batchin sovellustason kuittaus ei ole käytössä.

#### Send message Adapter Batch Acknowledgement (MCCI\_IN200102UV)

Tämän interaktion siirtokehys on ballot paketissa MCCI\_RM200100, joka ei sisällä acknowledgement-luokkaa..

Tämä interaktio on batch-sanoma, jonka vastaanottava adapteri lähettää käsiteltyään adapteritasolla batchin sisältämät interaktiot. Sisältää vastaanotetun batch-sanoman sisältämien interaktioiden Accept Ack-kuittaukset (vastaanottokuittaukset) eli vain MCCI\_IN000002UV01 interaktioita.

Adapteri voi tutkia interaktioiden acceptAckCode kentän arvot ja lähettää Accept Ack kuittaukset niiden perusteella tai kuittauskäytäntö voidaan sopia paikallisesti esim. kuittaus lähetetään vain virhetapauksessa ja tyhjä batch vastaus tarkoittaa, että kaikki sanomat on hyväksytty. Tyhjällä batch-sanomalla MCCI\_IN200102UV voidaan siis hoitaa koko batch-lähetyksen vastaanottokuittaus.

Ballot paketin mukaan adapteri saa torjua koko batch lähetyksen vain, jos batch sanoma on kokonaisuudessaan virheellinen esim. batchin sisältämien interaktioiden lukumäärä ei täsmää. Jos batch sisältää sekä virheellisiä että valideja interaktioita, validit on käsiteltävä normaalisti ja virheellisistä voidaan lähettää negatiivinen Accept Ack kuittaus MCCI\_IN200102UV interaktion sisällä.

Koko batchia tällä interaktiolla ei voi hylätä, koska se ei sisällä acknowledgement luokkaa vaan on käytettävä normaalia Accept Ack kuittausta.

### Suositeltu käyttötapa

Normaaleja sovellustason interaktioita kootaan bacth kehykseen ”Send Batch” (MCCI\_IN200100UV), esim. max 100 kpl. Sovitaan, että koko batchiin lähetetään yksi normaali vastaanottokuittaus ”Accept Ack” (MCCI\_IN000002UV01). Batchiä ei kuitata sovellustasolla response batchillä “Send Response Batch” (MCCI\_IN200101UV). Sovellustason vastaussanomat kootaan myös batchiin ”Send Batch” (MCCI\_IN200100UV) vastaavalla tavalla ja koko batch kuitataan normaalilla vastaanottokuittauksella MCCI\_IN000002UV01. Vastaussanomat voivat nyt tulla useana eri batchinä (ilman rajoituksia). (Täten säästytään problematiikasta tilanteessa, jossa osa sovellustason kuittauksista on negatiivisia).

Batchiä ei voi käyttää sellaisissa sovellusadaptereissa, joissa odotetaan edellisen sanoman kuittausta ennen kuin lähetetään uusi sanoma.

Sovellusaluekohtaisesti (esim. Kanta-palvelujen liikenne) määritellään, mitä interaktoita saa niputtaa samaan batchiin. Samoin sovellusaluekohtaisesti määritellään batchin sisältämien interaktioiden maksimimäärä.

## Vastaanottokuittaus - Accept Ack (MCCI\_IN000002UV01) Interaktio

Tämä interaktio on V3 2006 standardin mukainen normaali (adapteritason) vastaanottokuittaus.

Jokaisen vastaanottavan sovelluksen pitää osata lähettää vastaanottokuittaus (pyydettäessä) ja jokaisen lähettävän sovelluksen pitää pystyä vastaanottamaan vastaanottokuittaus. Ks. 2.3.1.9 acceptAckCode.

Vastaanotettuaan sanoman adapteri joko hyväksyy tai hylkää sanoman seuraavien seikkojen perusteella:

* Vastaanottavan järjestelmän tila
* Turvallisen tallennustilan saatavuus
* Sanoman syntaktinen oikeellisuus, jos tarkistukset tehdään tällä tasolla
* Adapteri tutkii saapuvasta sanomasta tarpeelliset kentät mm. interaktion id, type id, version, processing code

Adapteri tutkii myös siirtokehyksen acceptAckCode:sta onko vastaanottokuittaus käytössä. Jos sanoma voidaan hyväksyä ja kuittaus on aina voimassa adapteri lähettää lähettäjälle positiivisen vastaanottokuittauksen ”CA” koodilla ja siirtää sanoman sovelluksen käsiteltäväksi. Jos adapteri hylkää sanoman ja kuittaus on käytössä lähetetään vastaanottokuittaus ”CE” tai ”CR” koodilla ks. 2.3.1.9 ja 2.7.1.1.

Vastaanottokuittaukset lähetetään aina osoitteeseen, joka on ilmoitettu alkuperäisen sanoman siirtokehyksen elementissä sending device.

Jokaisen interaktion receiver responsibilities osioon kuuluu tavallaan interaktio MCCI\_IN000002UV01 vaikka sitä ei ole erikseen mainittu. Itse tämän interaktion siirtokehyksessä MCCI\_RM000200 (Message type MCCI\_MT000200UV01) acceptAckCode on vakioarvo ”NE”, joten sitä ei koskaan kuitata Accept Ackilla eli vastaanottokuittaukselle ei voi pyytää vastaanottokuittausta.

Alla interaktion kuvaavassa taulukossa siirtokehys (Transmission Wrapper) on MCCI\_MT000200UV01 ja kontrollikehys sekä varsinainen payload (Message Type) puuttuvat.

|  |  |
| --- | --- |
| Structured Name | Generic Message Transmission Definition Message Local Accept Acknowledgement |
| Trigger Event | Send Message Accept Acknowledgement  MCCI\_TE000002UV01 |
| Transmission Wrapper | Accept Acknowledgement MCCI\_MT000200UV01 |
| Control Act Wrapper |  |
| Message Type |  |
| Sending and Receiving Roles |  |
| Sender | Notification Message Sender MCCI\_AR900001UV01 |
| Receiver | Request Message Receiver MCCI\_AR900004UV01 |

Accept Ack kuittauksessa on keskeistä virheiden ilmoittaminen. Virheet kuvataan siirtokehyksen acknowledgement luokan avulla, joka on kuvattu kohdassa 2.3.2.1 ja alla kohdassa 2.7.1.1 on lisää asiasta.

### Vastaanottokuittaus synkronisessa/asynkronisessa liikenteessä

Asynkronisessa liikenteessä on syytä käyttää aina vastaanottokuittausta (myös positiivinen kuittaus) viestien perillemenon varmistamiseen, jos käytössä ei ole luotettavaa sanomanvälitystä esim. WS-RM (reliable messaging).

Synkronisessa request-response tyyppisessä liikenteessä, jossa normaalisti vastaussanomana on sovellustason vastaussanoma, vastaanottokuittausta voidaan käyttää vain virhetilanteissa ks. Synkroninen Web Services siirto 3.2.3.3.

Tiedostossa MCCI\_HD000200.htm on kuvattu taulukkomuodossa siirtokehyksen kentät.

Tiedostot MCCI\_IN000002UV01\_CA\_esim.xml, MCCI\_IN000002UV01\_CE\_esim.xml ja MCCI\_IN000002UV01\_CR\_esim.xml ovat esimerkkisanomia.

## Virheiden ilmoittaminen

### Virheiden ilmoittaminen siirtokehyksessä

#### Vastaanottokuittaus

Siirtokehyksessä ilmoitetaan kuitattavan sanoman teknisistä virheistä vastaanottokuittauksella (MCCI\_IN000002UV01 interaktio eli Accept Ack) Siirtokehyksen tyyppi on tässä interaktiossa MCCI\_MT000200UV01

Alla olevat taulukot on kopioitu kohdasta 2.3.2.1 acknowledgement

Vastaanottokuittauksen acknowledgement luokan elementti typeCode:

|  |  |  |
| --- | --- | --- |
| Koodi | tarkoitus | Suomennus |
| CA | Accept Acknowledgement Commit Accept | Sovellusadapteri on vastaanottanut sanoman onnistuneesti. |
| CE | Accept Acknowledgement Commit Error | Sovellusadapteri ei pystynyt käsittelemään sanomaa, uudelleenlähetystä kannattaa yrittää. |
| CR | Accept Acknowledgement Commit Reject | Kyseisen tyyppisen sanoman käsittely ei onnistu tässä sovelluksessa. Uudelleenlähetyksiä ei kannata yrittää. |

Mahdollisia virheitä: sanoman parserointi ei onnistu, osapuolitunnisteet (lähettäjä/vastaanottaja) ovat väärin, interaktio ei ole tuettu, profileId:n määrittämä profiili ei ole tuettu, templateId ei ole tuettu, standardin versio on väärä, processingCode on virheellinen, processingModeCode on virheellinen.

V3 standardin mukaan typeCode on ”CR” jos virhe on seuraavissa kentissä: interactionId, typeId, versionCode tai processingCode. Muista syistä johtuvissa virheissä koodi on ”CE”.

Virhe kuvataan yksityiskohtaisesti acknowledgementDetail luokan avulla. Alla kohdasta 2.3.2.6 kopioitu selitys.

Toistuvan acknowledgementDetail elementin avulla kerrotaan kuitattavasta sanomasta, mikä virhe oli kyseessä ja missä kohdassa. Tämä elementti vastaa 2.x:n tietoryhmää ERR. Elementin kaikki osat ovat myös vapaaehtoisia.

|  |  |  |  |
| --- | --- | --- | --- |
| elementti | 2.x  vastine | tietotyyppi | käyttötarkoitus/arvot |
| **TypeCode** | ERR-4 | CS | E=error, I=Information, W=Warning  Koodi kertoo virhetilanteen vakavuuden |
| **Code** | ERR-3+  ERR-5+  ERR-6 | CE | Virheilmoitus koodattuna(mielellään) tai pelkkänä tekstinä  koodisto AcknowledgementDetailCode (CWE) |
| **text** | ERR-7 | ED | Virheeseen liittyvä lisätieto |
| **Location** | ERR-2 | SET(ST) | Virheen sijainti |

Code elementin koodisto on AcknowledgementDetailCode (2.16.840.1.113883.11.19638). Koodisto on CWE eli omienkin koodistojen käyttö on sallittu, mutta ym. koodisto sisältää koodit tavallisimpiin virhetilanteisiin.

esim:

<acknowledgement>

<typeCode code="CE"/>

<!-- kuitattavan sanoman id -->

<targetMessage>

<id root="1.2.246.10.6280613.18.2004.225. 2006.23651"/>

</targetMessage>

<!-- Virheen kuvaus acknoledgementDetail elementissä -->

<acknowledgementDetail>

<!-- typeCode ilmoittaa virheen vakavuuden, "E"=error "W"=warning -->

<typeCode code="E"/>

<!-- virheen koodi tai virhe tekstinä code attribuuutissa -->

<code code="RTUDEST" codeSystem="2.16.840.1.113883.11.19638" codeSystemName="AcknowledgementDetailCode" displayName="Unknown destination"/>

<text>mahdollinen virheeseen liittyvä lisätieto</text>

<location>mahdollinen virheen paikan kuvaus</location>

</acknowledgementDetail>

</acknowledgement>

#### Sovellustason kuittaus

Sovellustason vastaussanomissa (kuittauksissa) siirtokehys on tyyppiä MCCI\_MT000300UV01, joka sisältää myös acknowledgement luokan ja tällöin typeCode saa seuraavia arvoja:

Sovellustason kuittaukset:

|  |  |  |
| --- | --- | --- |
| Koodi | tarkoitus | Suomennus |
| AA | Application Acknowledgement Accept | Sovellustason OK kuittaus. |
| AE | Application Acknowledgement Error | Sovellustason virhe. (Sanomaa ei kannata lähettää uudestaan). |
| AR | Application Acknowledgement Reject | Sanomakäsittely ei onnistunut. (Lähettäjä tekee parametroidun määrän uudelleenlähetyksiä) |

Sovellustason virheitä ei kuvata tarkemmin siirtokehyksessä acknowledgementDetail luokan avulla vaan kontrollikehyksessä reasonOf rakenteessa (alla 2.7.2), koska ne liittyvät sovellustason prosessiin. Jos kuitenkin on kyseessä tekninen virhe, käytetään luokkaa acknowledgementDetail.

MCCI\_MT000200UV01 R-MIM on tiedostossa MCCI\_RM000200UV01.htm ja taulukkomuodossa tiedostossa MCCI\_HD000200UV01.htm

MCCI\_MT000300UV01 R-MIM on tiedostossa MCCI\_RM000300UV01.htm ja taulukkomuodossa tiedostossa MCCI\_HD000300UV01.htm

### Sovellustason virheet kontrollikehyksessä

Sovellustason kuittaussanomissa välitetään siirtokehyksessä MCCI\_MT000300UV01 sovellustason kuittauskoodi edellisessä kohdassa (2.7.1.2) kuvatulla tavalla.

Sovellustason virheet liittyvät sanoman tietosisältöön ja sen käsittelyyn sovellustasolla.

Virhetilanteissa (AE ja AR) virheen tarkempi kuvaus sijoitetaan kontrollikehyksen **reasonOf** rakenteeseen, joka on kuvattu kohdassa 2.3.3.12. Virheet ilmoitetaan DetectedissueEvent luokan attribuuteissa. Mahdolliset virheet ja DetectedIssueEvent luokassa käytettävä koodisto on kuvattava implementointioppaassa.

Virheitä voitaisiin kuvata myös kontrollikehyksen reasonCode (koodisto actReason CWE) elementin avulla. ActReason koodistossa on joitakin arvoja, joita voidaan käyttää virhetilanteissa mutta tällöin reasonOf rakenteen on oltava tyhjä. Jos taas käytetään reasonOf rakennetta reasonCode on oltava tyhjä.

ReasonOf rakenne on suositeltava tapa sovellustason virheiden ilmoittamiseen.

Control Act R-MIM on tiedostossa MCAI\_RM700200UV01.png ja taulukkomuodossa tiedostossa MCAI\_HD700200UV01.htm

DetectedIssueEvent R-MIM on tiedostossa MCAI\_RM900000.png ja taulukkomuodossa tiedostossa MCAI\_HD900000.htm.

Alla taulukkomuodossa 2.3.3.12 kohdassa kuvatut DetectedIssueEvent luokan attribuutit. (taulukko ”Ajanvarausrajapinnat - Tekninen liittymämäärittely” dokumentista)

|  |  |  |
| --- | --- | --- |
| **Elementti** | **Pakollisuus** | **Selite** |
| ClassCode | 1..1 | vakioarvo "ALRT" |
| MoodCode | 1..1 | vakioarvo "EVN" |
| id (II) | 0..1 | virheen yksikäsitteinen tunniste |
| code (CD) | 1..1 | virhe tyyppi koodatussa muodossa |
| text(ST) | 0..1 | virheen lisätiedot |
| value (ANY) | 0..1 | tarkka virhetieto (Observation value-tyyppisesti). |

DetectedIssueEvent luokan käyttö ja code elementissä käytettävä koodisto on kuvattava implementointioppaassa.

Code elementti on pakollinen. Jos koodistoa ei ole, tässä kentässä voidaan käyttää koodistoa ActDetectedIssueCode code="1" codeSystem="1.2.246.537.5.40302" displayName="Issue Detected" ja ilmoittaa tarkka virhetieto value elementissä tekstinä.

Alla esimerkkinä ”Ajanvarausrajapinnat - Tekninen liittymämäärittely” dokumentissa kuvattu koodisto sovellustason virheiden ilmoittamiseen.

**Esim: Ajanvarauksen sovellustason virhekoodit.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **DisplayName** | **detectedIssueEvent.text (esimerkiksi)** | **Interaktiot, joihin voi tulla vastauksena** |
| S01 | Yleinen virhe | Virhe pyynnön käsittelyssä | pyyntö, kysely |
| S02 | Puuttuva tieto | Pakollinen tieto (palvelutuote) puuttuu | pyyntö, kysely |
| S03 | Virheellinen koodi | Palvelutuotetta "1" ei löydy ajanvarausjärjestelmästä | pyyntö, kysely |
| S04 | Virheellinen koodisto | Tuntematon palvelutuotteiden koodisto | pyyntö, kysely |
| S05 | Virheellinen tieto | Haettu aikaväli on virheellinen | pyyntö, kysely |
| S06 | Liikaa pyydettyjä tietoja | Kyselyyn ei voida palauttaa pyydettyä määrää tietoja | kysely |
| S10 | Asiakasta ei löydy | Henkilöä ei löydy ajanvarausjärjestelmästä | pyyntö, varattujen aikojen kysely |
| S11 | Varattava aika ei ole vapaana | Pyydettyyn aikaan ei ole saatavilla vapaita aikoja | pyyntö |
| S12 | Resursseja ei vapaana | Pyydettyyn palveluun tarvittavia resursseja ei ole saatavilla | pyyntö |
| S13 | Suostumusvirhe | Tarvittu suostumus puuttuu tai on virheellinen | varattujen aikojen kysely |
| S14 | Ajanvarauspyyntö hylätty | Ajanvarauksen perumispyyntö hylätty | pyyntö |

XML esimerkki sovellustason virheestä: (ajanvaraus)

<PRSC\_IN040203FI01>

<id .../>

...

<receiver .../>

<sender .../>

<acknowledgement>

<!-- Sovellustason virhe -->

<typeCode code="AE"/>

<targetMessage>

<id root="1.2.246.10.2458998.11.2006.19"/>

</targetMessage>

</acknowledgement>

<controlActProcess moodCode="EVN">

<reasonOf>

<detectedIssueEvent>

<code code="S01" codeSystem="1.2.246.537.5.40204"/>

<text mediaType="text/plain">Tuntematon ajanvaraus</text>

</detectedIssueEvent>

</reasonOf>

</controlActProcess>

</PRSC\_IN040203FI01>

# Tiedonsiirto

## Yleiset periaatteet

### perusperiaatteet



Kuva 21: Sovellusarkkitehtuuri

Lähde: HL7 Version 3 Standard: Abstract transport specification, September 2005 Ballot

HL7:n tiedonsiirron yleisiä (teknologiasta riippumattomia) periaatteita kuvataan määrityksessä HL7 Version 3 Standard: Abstract transport specification.

Ylimmällä sovellustasolla (application layer) sijaitsee:

* Varsinainen HL7-sovellus (application)
* adapterit (messaginng adapters), jotka eristävät sovelluksen sanomanvälitysinfrastruktuurista. Adapterit osaavat HL7-sovellustason protokollat ja muodostavat liitännän siirtoprotokollatasoon

Seuraavalla tasolla ovat siirtoprotokolla-adapterit (messaging infrastructure), jotka eivät tiedä HL7-protokollista mitään, mutta osaavat tietyn siirtoprotokollaan, esim. web services.

Seuraavalla tasolla sijaitsevat siirtotekniikka-adapterit, joiden avulla siirtoprotokollataso liittyy varsinaiseen siirtotekniikkaan, esim. http.



Kuva 22: Sanomanvälityksen abstraktointitasot

Lähde: HL7 Version 3 Standard: Abstract transport specification, May 2007 Ballot

Kerrosajattelun avulla kuljetusprotokolla ja varsinainen siirtotason toteutus pystytään pitämään erillään (eristämään) toisistaan.

Määrityksissä lähettäjä (sender) ja vastaanottaja (receiver) ovat sovellustasolla olevia adaptereita ja sanoman lähde (source) ja kohde (destination) ovat siirtoprotokollatasolla olevia adaptereita.

Siirto voidaan toteuttaa normaalisiirtona, jolloin lähettäjä lähettää sanoman. HL7-määrittelee myös pollaukseen perustuvan siirron, jossa vastaanottaja käy noutamassa sanomat lähettäjän jonosta. Tätä siirtotapaa ei käsitellä tarkemmin tässä dokumentissa.

Abstrakti tiedonsiirtomääritys sisältää useita vaatimuksia, joita siirtoprotokollatason tulee toteuttaa. Niitä kaikkia ei käydä tässä läpi, sillä ne on tarkoitettu teknisille komiteoille, jotka toteuttavat siirtoprotokollatasoon sidottuja määrityksiä.

### Siirron luotettavuus

#### Siirtoprotokollataso – commit ack

Siirtoprotokollatason määrityksessä tulee kuvata, kuinka luotettava tiedonvälitys kahden pisteen välillä (source-destination) voidaan toteuttaa (ilman HL7:n sovellustason protokollia). HL7:n sovellustasolla on käytössä sovellustason kuittaus (vastaussanoma) ja adapteritason vastaanottokuittaus (Accept Ack). Siirtoprotokollatasolla puhutaan commit ack:istä, joka lähetetään sen jälkeen, kun siirrettävä aineisto on vastaanotettu protokollatasolla ja turvallisesti talletettu eteenpäin välitystä varten. Commit ACK:illa ei ole HL7:n kanssa varsinaista tekemistä V3 messaging määrityksissä, sillä se kulkee vain siirtoprotokollatason adaptereiden (source-destination) välillä. HL7 versiossa 2 sen sijaan commit ack on oleellinen osa siirtoprotokollaa.

Siirtoprotokollatasolla taataan siis sanoman perillemeno (tai ilmoitus perille menemättömyydestä), mutta duplikaattiesto kuuluu sovellustason adapterille. Siirtoprotokollatason vaatimuksiin kuuluu nyt myös sanomien järjestyksen säilyttäminen, vaikkakin myös sovellusadapteritasolla on järjestyksen takaava protokolla (sequence number protocol).

Suomessa on CDA-dokumenttien siirrossa käytetty SOAP-tyyppistä tiedonsiirtoa ja kuittausta HL7Ficommit. Koska V3 messaging-käytössä on tarkoitus siirtyä mahdollisimman suureen kv. yhteensopivuuteen ja todellisen SOAP:in käyttöön, ei HL7Ficommit-kuittausta voida enää käyttää. Negatiivinen kuittaus saadaan SOAP-standardin mukaisena SOAP fault-sanomana tai ei-OK http-kuittauskoodina. Positiiviseksi kuittaukseksi tulkitaan OK-http-koodit (2XX, esim. 200 tai 202).

Jos siirtokanavaa ei pidetä tarpeeksi luotettavana, voidaan käyttää ylemmän tason luotettavan tiedonsiirron protokollaa: WS-ReliableMessaging ( ks. kappale 3.2.6).

#### Sovellustason adapterit ja vastaanottokuittaus

Sovellustason adapteri erottaa sovelluksen siirtoprotokollatasosta. Sovellustason adaptereiden välillä (sender-receiver) kulkee vastaanottokuittaus (MCCI\_IN000002UV01 = Accept Ack), jos sellaista on siirtokehyksessä (transmission wrapper) pyydetty (acceptAckCode). Vastaanottokuittaus voidaan pyytää aina (”AL”), vain virhetilanteissa (”ER”) tai ei koskaan (”NE”). (Joissakin tilanteissa voidaan sanoa, että asynkronisessa liikenteessä riittää, että palvelin vastaa HTTP - koodeilla 200 tai 202. Tällöin viesti on mennyt ainakin perille, vaikkakaan sitä ei ole välttämättä ymmärretty tai sitä ei käsitellä sovellustasolla.)

Sovellustason interaktioissa määritellään, mitä sovellustason kuittausta käytetään, eikä se kuulu sovellustason adapterille. Interaktion receiver responsibilities osiossa määritetään millä interaktiolla tai interaktioilla ko. interaktioon on vastattava ja vastausinteraktiossa välitettävä sanoma on sovellustason kuittaussanoma. Yleensä on kyseessä pyyntö ja sen vastauksena/kuittauksena hyväksyvä tai hylkäävä interaktio.

Sovellustason adapterilla pitää olla tietämys siirtokehyksestä sekä siinä esitetystä sovellustason protokollasta (esim. batch, sequence number protocol). Adapteri voi välittää sanoman sovellustasolle merkkijonona, mutta se voi myös purkaa sanoman objektimuotoon. Validointi voidaan tehdä adapterissa tai varsinaisella sovellustasolla.

Adapteritason vastaanottokuittauksessa (MCCI\_IN000002UV01) ei ole kontrollikehystä eikä varsinaista payloadia. Kuittaustiedot sijoitetaan siirtokehyksen elementtiin acknowledgement. Negatiivinen vastaanottokuittaus voidaan lähettää esim. seuraavissa tapauksissa:

* vastaanottavaa sovellusta ei tunnisteta tai lähettäjä ei ole sallittu
* interaktiota ei tueta
* sanoma ei noudata sovittua profiilia

Kuittaussanomassa kuittaustiedot sijoitetaan siirtokehykseen. Accept Ack kuittaus on kuvattu kohdassa 2.6.

Koska adapteri ei ole tietoinen sovellustason interaktioista, esim. kyselysanomat ja normaalit päivityssanomat käsitellään samalla tavalla adapteritasolla. Liikenne voi olla synkronista tai asynkronista. Adapteri palauttaa sovellustasolle lähetyspyyntöön negatiivisen vastineen, jos:

* protokollatason adapteri ilmoittaa vastaanottopäästä tulleesta negatiivisesta commit-tason kuittauksesta
* Lähetys ei onnistunut uudelleenyritystenkään jälkeen
* Pyydettyä siirtotapaa ei tueta

#### Gateway

Gateway on järjestelmä, joka toteuttaa businesstason logiikkaa muiden sovellusten puolesta. Lähettäjän kannalta gateway on normaali HL7-sovellus. Gateway kuitenkin välittää sanoman eteenpäin ja yleensä muodostaa sanoman uudestaan, jolloin myös message.id muuttuu. Gateway voi suorittaa esim. täydellisen protokollamuunnoksen tai generoida yhdestä interaktiosta useita interaktioita. Gateway on erittäin hyödyllinen esim. tapauksessa, jossa potilastietojen master-järjestelmä haluaa ilmoittaa usealle erillisjärjestelmälle potilaan sisäänkirjauksesta. Tällöin sisäänkirjaussanoma lähetetään gatewaylle, joka rakentaa alkuperäisestä sanomasta sanomat kullekin erillisjärjestelmälle ja hoitaa noista järjestelmistä tulleet sovellustason kuittaukset. Gatewayn avulla voidaan vähentää keskeisten järjestelmien liittyminen lukumäärää ja yksinkertaistaa business-logiikkaa.

Koska gateway on normaali HL7-sovellus, kuittauskäytännössä lähettäjän ja gatewayn välillä ei ole tulkintaongelmia.

#### Bridge - silta

Bridgen tehtävänä on sanomien reititys ja yleensä protokollamuutokset siirtoprotokollatasolla. Muunnos voi olla esim. web services – MLLP. Bridge ei ole HL7-sovellus ja sanoman sisältöön (HL7 composite message) ei kosketa. Bridge lähettää alkuperäiselle lähettäjälle commit ack:in, kun sanoma on vastaanotettu ja talletettu. Ongelma syntyy, jos bridge ei saakaan välitettyä sanomaa eteenpäin. Tällöin bridgen pitää lähettää negatiivinen vastaanottokuittaus (MCCI\_IN000002UV01), jos vastaanottokuittausta on sanomassa pyydetty. Tämän vuoksi myös bridgen pitää ymmärtää siirtokehyksen rakenne (transmission wrapper).

HL7-sovellusten yleissääntö: Jos siirtokehyksessä on eri vastaanottaja kuin sovellus itse, on sovellusta pyydetty olemaan siltana ja välittämään sanoma eteenpäin todelliselle vastaanottajalle. Jos tämä ei onnistu, pitää lähettää negatiivinen vastaanottokuittaus.



Kuva 23: Silta siirtotason protokollamuuntimena

Lähde: HL7 Version 3 Standard: Abstract transport specification, Mayr 2007 Ballot

#### Reititin (intermediary, router)

Reititin toimii kuljetustasolla. Se voi reitittää sanomia ja muuntaa kuljetustason protokollaa, kunhan siirtotason protokolla pysyy samana. Välittimellä on lupa muuttaa siirtoprotokollatason metatietoja. Välitin ei lähetä mitään ”ylemmän tason” kuittauksia, sillä commit ack:kikin on määritelty siirtoprotokollatasolle. Reititin ei siis mitenkään näy HL7-sanomaliikenteessä.



Kuva 24: Gatewayn, sillan ja reitittimen ero

Lähde: HL7 Version 3 Standard: Abstract transport specification, May 2007 Ballot

### Osoitteistus

Siirtoprotokollatason määrityksissä pitää kuvata, kuinka siirtokehyksessä (transmission wrapper) ilmoitetut looginen lähettäjä ja vastaanottaja kuvataan siirtotasolla fyysisiksi osoitteiksi.

### Yhteenvetokuva

Adapteri (sovellustasolla), joka voi käyttää normaalilähetyksen lisäksi polling- mekanismia. Voi toteuttaa luotetun siirron sequence number protokollalla.

varsinainen sovellus

interaktion mukainen varsinainen sanomasisältö

semanttinen sisältö

Kontrollikehys, joka riippuu siitä, onko kyseessä normaali sanoma, query, vai master file/registry

interaktio

määrittelee

tarvitsee

sanoma-kehyksen

tietoja

Siirtokehys, johon vaikuttaa, onko kyseessä yksittäinen sanoma vai batch

Siirtoprotokolla (esim. Web services/SOAP). Voi toteuttaa luotettavan siirron esim. WS RM-standardilla

osoite

loogisesta

fyysiseksi

Kuljetusprotokolla (esim. http)

Kuva 25 – kerrosajattelun yhteenveto

## Web services tiedonsiirto

Tässä dokumentissa oletetaan, että lukija on aiemmin perehtynyt web services – tekniikoihin, WS-I basic profileen sekä WSDL-kuvauskieleen. Implementointia tekevän pitää lukea myös alkuperäinen HL7 web services profile – dokumentti (”Transport Specification - Web Services Profile, Release 2” May 2007 HL7-äänestyspaketti ja jatkossa normatiivinen versio). Lyhenteitä ja termejä on selitetty liitetiedostossa Wslyhe.doc.

### Web services yleiskuvaus

Web services tekniikoilla tiedonsiirto saadaan toteutettua alustariippumattomasti.



Kuva 26 - Web services standardiperheen yleisrakenne:

Lähde:

HL7 Version 3 standard – Transport specification – Web services profile, release 2

Alimmalla kuljetustasolla tyypillisin protokolla on http tai http/s. Varsinaisella siirtotasolla käytetään XML:ää siirron aikaisena esitysmuotona sekä SOAP-protokollaa. SOAP jakaa siirrettävän aineiston bodyyn, jossa on varsinainen payload sekä headeriiin, joka sisältää ohjaustietoja. SOAP:in päälle on web services standardiperheessä rakennettu useita ylemmän tason protokollia, jotka käytännössä perustuvat SOAP-headeriin sijoitettaviin ohjaustietoihin. Messaging tasolla määritellään perussanomanvälitys. Web services käyttö dokumentoidaan WSDL-kuvauskielen avulla, joskin muutakin metadataa tarvitaan.

### HL7:n web services profiilit

HL7 on määritellyt web services standardeille soveltamisohjeita. Sen nimi on HL7WSP (health level 7 web services profiles). Soveltamisohjeiden pohjana on WS-I basic profile v 1.0. Itse asiassa profiileita on useita:

* Basic profile – perusprofiili
* Addressing profile – osoitteistusprofiili
* Security profile – tietoturvaprofiili
* Reliable messaging – luotettavan sanomavälityksen profiili

Määräykset on koodattu seuraavasti:

HL7-WS[P|A|S|RM]nnn ohje

Kirjainten selitys on seuraava: P = basic profile, A = addressing profile, S= security profile, RM = reliable messaging profile.

HL7 Finland on määritellyt, että web services osapuolten pitää tukea perusprofiilia sekä osoitteistusprofiilia sekä tässä dokumentissa mainittuja lisärajoituksia.

Poikkeuksena sallitaan synkronisessa web services liikenteessä poikkeaminen perusprofiilin kohdasta **HL7-WEP100** jonka mukaan SOAP Bodyn juurielementti (”metodin nimi”) on aina interaktion tunnus koska usean vaihtoehtoisen interaktion käyttö paluuviestinä ei onnistu muutoin.

### Perusprofiili (HL7 Web services basic profile, HL7WSP)

Siirrettäessä HL7 Version 3 sanomia web services tyylisesti, on SOAP-Bodyn rakenne seuraava, oli kyseessä sitten varsinainen sanoma tai sovellustason kuittaussanoma:



Kuva 27 – Web services siirron rakenne

HL7 Version 3 sanomasta käytetään termiä HL7 V3 composite message, koska sanoma itse asiassa koostuu kolmesta osasta: siirtokehys eli transmission wrapper, kontrollikehys eli control act wrapper ja varsinainen sanoma. Siirtokehys ja kontrollikehys ovat kaikille sanomille lähes yhteisiä rakenteita. Ne kuitenkin erikseen määrätään jokaisessa interaktiossa. Vastaanottokuittauksesta (Accept ACK, Message adapter ACK) kontrollikehys puuttuu. Vastaanottokuittaus lähetetään, jos lähettäjä sellaista pyytää (siirtokehyksessä). Siirtokehys vastaa 2.x sanomamääritysten MSH-segmenttiä ja kontrollikehys 2.x:n EVN-segmenttiä.

.

#### SOAP-body

Web services tyylinen soveltamistapa paketoi HL7 Composite messagen metodin nimen alle. Metodin nimi korvaa siirtokehyksen message-elementin.

Metodin nimi on sama kuin interaktion tunnus. Tämä on määräys **HL7-WSP100.**

Lisäksi määritellään namespace: xmlns=”urn:hl7-org:v3”

Esimerkki:

<soap:Envelope xmlns:soap="...">

<soap:Body>

<PRPA\_IN101201 xmlns="urn:hl7-org:v3">

<id root="2.16.840.1.113883.9876.393.79"/>

<creationTime value="20040205132400"/>

<versionCode code="V3-2006"/>

...

</PRPA\_IN101201>

</soap:Body>

</soap:Envelope>

HL7FI Poikkeus:

Synkronisessa web services liikenteessä sallitaan poikkeaminen tästä säännöstä request-response tyyppisessä liikenteessä paluuviestin osalta, koska usean vaihtoehtoisen interaktion käyttö paluuviestinä ei onnistu muutoin. (WSDL sallii message määrittelyissä vain yhden output sanoman.)

#### WSDL-kuvaukset

Perusohjeet

WSDL (web services description language) on XML-kieli, jolla kuvataan palveluosapuolen rajapinta XML-tekniikalla. Yleensä adapterisovellukset rakentavat liikennöinnissä tarvittavan WS-rajapinnan automaattisesti vastapään sovelluksen WSDL:n avulla. WSDL on siis koneellisesti luettava ”konfigurointiohje”. Se sitoo varsinaisen sanomamäärityksen operaatioihin ja palveluportteihin.

WSDL:n kuvaamaan rajapintaan vaikuttaa liikenteen tyyppi (interaktio). Perusliikennetyyppejä ovat:

* yksisuuntainen liikenne (one-way)
* (synkroninen) kysely-vastausliikenne (request-response)
* asynkroninen vastaus (solicit response)
* ilmoitus (notification)



Kuva 28 –perusliikennetyypit (MEP)

Liikenteen teknisen toteutuksen kannalta tapaukset 2 ja 3 ovat keskenään samanlaisia sekä 1 ja 4. Perusliikennetyypin englanninkielinen termi on Message Exchange Pattern, MEP.

**HL7-WSP200**: WSDL-dokumentin tulee perustua sovelluksen rooliin.

**HL7-WSP201**: Attribuutin /wsdl:definitions/@name tulee olla sovelluksen roolille annettu tunnus.

Seuraavassa taulukossa {NAME} tarkoittaa sovelluksen roolia. Seuraava taulukko ei ole määräys, vaan ehdotus hyväksi tavaksi.

| WSDL Artifact | Proposed Naming |
| --- | --- |
| message | {Interaction Artifact Id}\_Message |
| portType | {NAME}\_PortType |
| operation | {NAME}\_{Interaction Artifact Id} |
| SOAP 1.1 binding | {NAME}\_Binding |
| SOAP 1.2 binding | {NAME}\_BindingSoap12 |
| SOAP 1.1 port | {NAME}\_Port |
| SOAP 1.2 port | {NAME}\_PortSoap12 |

Esimerkki, jossa sovelluksen rooli on PRPA\_AR101002:

For wsdl:definitions/@name="PRPA\_AR101002":

message -> "PRPA\_IN101001\_Message"

portType -> "PRPA\_AR101002\_PortType"

operation -> "PRPA\_AR101002\_PRPA\_IN101001"

SOAP 1.1 binding -> "PRPA\_AR101002\_Binding"

SOAP 1.2 binding -> "PRPA\_AR101002\_Binding\_Soap12"

SOAP 1.1 port -> "PRPA\_AR101002\_Port"

SOAP 1.2 port -> "PRPA\_AR101002\_PortSoap12"

HL7 määrittelee päästä päähän interaktioita ja WSDL siis sovelluksen rooliin perustuvan rajapinnan. Täten tietyn interaktion kokonaisuus näkyy kahdesta WSDL:stä, lähettäjän ja vastaanottajan. Tietyssä WSDL:lässä voi olla usean interaktion (toisen pään) määrittely.

**HL7-WSP202**: WSDL:n targetNamespace-elementin arvon on oltava: ”urn:hl7-org:v3”

WSDL types

**HL7-WSP203** : Jokaisen tuettavan HL7-interaktion schema pitää includoida types-osioon.

**HL7-WSP204:** Jos kuittaussanomat määritellään WSDL:lässä, niin niiden schemat pitää includoida types –osioon ja niille pitää määritellä root-elementti, jolle annetaan nimeksi kuittauksen tunnus.

Jälkimmäinen vaatimus johtuu siitä, että interaktioschemoissa on määritelty valmiiksi juurielementti interaktion nimen mukaisesti, mutta kuittaussanomissa tätä ei ole tehty, sillä ne ovat yleiskäyttöisiä.

Esimerkki:

<types>

<xsd:schema

elementFormDefault="qualified"

targetNamespace="urn:hl7-org:v3"

xmlns:tns="urn:hl7-org:v3">

<xsd:include schemaLocation="MCCI\_MT000200.xsd"/>

<xsd:include schemaLocation="MCCI\_MT000300.xsd"/>

<xsd:include schemaLocation="PRPA\_IN101001.xsd"/>

<xsd:element

name="MCCI\_MT000200"

type="tns:MCCI\_MT000200.Message"/>

<xsd:element

name="MCCI\_MT000300"

type="tns:MCCI\_MT000300.Message"/>

</xsd:schema>

</types>

WSDL message

**HL7-WSP205:** Yhdensuuntaiselle interaktiolle tehdään yksi message-määritys ja kysymys-vastaus-tyyppisille interaktioille kaksi message-määritystä.

**HL7-WSP206:** Message määrittelyssä pitää olla part-elementti, jonka nimi on ”Body” ja joka viittaa interaktion juurielementtiin.

Tässä on siis kyseessä vain 1-1 mappaus types-osioon.

WSDL portType

Sanomat ja operaatiot sidotaan toisiinsa portType-määrittelyllä.

**HL7-WSP207:** Jokaiselle input- ja output-sanomalle pitää määritellä attribuutti wsa:action.

**HL7-WSP208:** Edellisen kohdan wsa:action arvo on interaktion tunnus: wsdl:operation/wsdl:input/ @wsa:Action=”urn:hl7-org:v3:{Interaction\_Artifact\_Id}”. HL7 Finland lieventää tätä vaatimusta siten, että @wsa:Action kentän sisältönä on edellä määritelty tietosisältö, mutta ei välttämättä samassa muodossa.

**HL7-WSP209:** Adaperikuittauksille eli vastaanottokuittauksille käytetään arvoa: wsdl:operation/wsdl:input/ @wsa:Action=”urn:hl7-org:v3:AcceptAcknowledgement”.

HL7 Finland lieventää tätä vaatimusta siten, että @wsa:Action kentän sisältönä on edellä määritelty tietosisältö, mutta ei välttämättä samassa muodossa.

**HL7-WSP210:** Sovellustason kuittauksille käytetään arvoa: wsdl:operation/wsdl:input/ @wsa:Action=”urn:hl7-org:v3:ApplicationAcknowledgement”.

HL7 Finland lieventää tätä vaatimusta siten, että @wsa:Action kentän sisältönä on edellä määritelty tietosisältö, mutta ei välttämättä samassa muodossa.

Esimerkki:

<message name="PRPA\_IN101001\_Message">

<part name="Body" element="hl7:PRPA\_IN101001"/>

</message>

<message name="MCCI\_MT000200\_Message">

<part name="Body" element="hl7:MCCI\_MT000200"/>

</message>

<message name="MCCI\_MT000300\_Message">

<part name="Body" element="hl7:MCCI\_MT000300"/>

</message>

<portType name="PRPA\_AR101002\_PortType">

<operation name="PRPA\_AR101002\_PRPA\_IN101001">

<input message="hl7:PRPA\_IN101001\_Message"

wsa:Action="urn:hl7-org:v3:PRPA\_IN101001"/>

<output message="hl7:MCCI\_MT000200\_Message"

wsa:Action="urn:hl7-org:v3:MCCI\_MT000200"/>

</operation>

<operation name="PRPA\_AR101002\_AcceptAcknowledgement">

<input message="hl7:MCCI\_MT000200\_Message"

wsa:Action="urn:hl7-org:v3:AcceptAcknowledgement"/>

</operation>

<operation name="PRPA\_AR101002\_ApplicationAcknowledgement">

<input message="hl7:MCCI\_MT000300\_Message"

wsa:Action="urn:hl7-org:v3:ApplicationAcknowledgement"/>

</operation>

</portType>

SOAP-esimerkki actionista:

<soap:Header>

…

<wsa:Action>urn:hl7-org:v3:PRPA\_IN101201</wsa:Action>

….

</soap:Header>

**WSDL binding (WSDL 1.1)**

Bindingin avulla operaatiot sidotaan siirtotekniikkaan. Sidoksia voi olla useisiin tekniikoihin, mutta suositeltavin on http ja http/s.

**HL7-WSP211:** Bindingissa jokaiselle operaatiolle pitää määritellä soapAction (soap:operation/@soapAction). Sen pitää olla sama kuin wsa:action portType-osiossa.

**HL7-WSP212:** HL7 WSDL:lässä pitää käyttää SOAP bindingia, joka on määritelty WSDL 1.1 määrityksen kappaleessa 3.

**HL7-WSP213:** HL7 SOAP-sanomissa pitää käyttää document/literal – tyyliä.

**HL7-WSP214:** SOAP 1.2 bindingissä pitää olla määritelty: soap12:operation/@soapActionRequired=”true”.

Esimerkki (SOAP 1.1 ja SOAP 1.2) bindingistä:

<binding name="PRPA\_AR101002\_Binding"

type="hl7:PRPA\_AR101002\_PortType">

<soap:binding style="document"

transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="PRPA\_AR101002\_PRPA\_IN101001">

<soap:operation

soapAction="urn:hl7-org:v3:PRPA\_IN101001"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

<operation name="PRPA\_AR101002\_AcceptAcknowledgement">

<soap:operation

soapAction="urn:hl7-org:v3:AcceptAcknowledgement"/>

<input>

<soap:body use="literal"/>

</input>

</operation>

<operation name="PRPA\_AR101002\_ApplicationAcknowledgement">

<soap:operation

soapAction="urn:hl7-org:v3:ApplicationAcknowledgement"/>

<input>

<soap:body use="literal"/>

</input>

</operation>

</binding>

<binding name="PRPA\_AR101002\_Binding\_Soap12"

type="hl7:PRPA\_AR101002\_PortType">

<soap12:binding style="document"

transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="PRPA\_AR101002\_PRPA\_IN101001">

<soap12:operation

soapAction="urn:hl7-org:v3:PRPA\_IN101001"

soapActionRequired="true"/>

<input>

<soap12:body use="literal"/>

</input>

<output>

<soap12:body use="literal"/>

</output>

</operation>

<operation name="PRPA\_AR101002\_AcceptAcknowledgement">

<soap12:operation

soapAction="urn:hl7-org:v3:AcceptAcknowledgement"

soapActionRequired="true"/>

<input>

<soap12:body use="literal"/>

</input>

</operation>

<operation name="PRPA\_AR101002\_ApplicationAcknowledgement">

<soap12:operation

soapAction="urn:hl7-org:v3:ApplicationAcknowledgement"

soapActionRequired="true"/>

<input>

<soap12:body use="literal"/>

</input>

</operation>

</binding>

WSDL service

Lopuksi nimetään palvelut, sidotaan ne binding-määrittelyihin ja kerrotaan missä portissa ja missä osoitteessa palvelu on saatavissa.

Esimerkki:

<service name="PRPA\_AR101002\_Service">

<port

name="PRPA\_AR101002\_Port"

binding="hl7:PRPA\_AR101002\_Binding">

<soap:address

location="http://servicelocation/PRPA\_AR101002"/>

</port>

<port

name="PRPA\_AR101002\_PortSoap12"

binding="hl7:PRPA\_AR101002\_Binding\_Soap12">

<soap12:address

location="http://servicelocation/PRPA\_AR101002"/>

</port>

</service>

#### Synkroninen Web Services siirto

HL7 V3 interaktiossa sanomaan voi tulla useita vaihtoehtoisia vastauksia ja lisäksi vastaanottokuittaus (Accept Ack). WSDL määritys ei salli operaatiossa sanomaan kuin yhden vastaussanoman ja perusprofiilin kohta HL7-WSP100 vaatii, että SOAP Bodyn ensimmäisen elementin on oltava interaktion tunnus. HL7 Web Services perusprofiili on käytännössä tehty asynkronisen liikenteen määrittelyyn.

Asynkronisen liikenteen toteuttaminen vaatii, että molemmat osapuolet voivat toimia server moodissa. Kaikille clienteille server moodi ei kuitenkaan ole mahdollinen.

Tämän vuoksi **HL7 Finland sallii poikkeamisen perusprofiilin kohdasta HL7-WSP100 synkronisen Web Services liikenteen vastausinteraktioiden toteuttamisessa request-response tyyppisessä liikenteessä**. Kansainvälisissä toteutuksissa (Hollanti, Kanada) on toimittu samalla tavalla eikä HL7 web services profiilien dokumentaatio anna ohjeita tämän ongelman ratkaisemiseksi.

Synkronisessa liikenteessä positiivista vastaanottokuittausta ei voi käyttää onnistuneen pyynnön jälkeen, koska tässä tapauksessa palautetaan sovellustason vastaussanoma, eli jos kuittausta käytetään pyyntösanomassa acceptAckCode code="ER". Palautettavassa vastausinteraktiossa ei voi pyytää vastaanottokuittausta eli acceptAckCode code="NE".

HL7 Finlandin suositus vastausinteraktioiden toteuttamiseksi synkronisessa Web Services liikenteessä:

Vastaanottokuittausta pyydetään vain virhetilanteessa <acceptAckCode code="ER"/>.

Vastaussanoman juurielementin nimeämisessä ja WSDL kuvauksessa käytetään seuraavaa nimeämistapaa ja rakennetta:

* Vastaussanoman juurielementti nimetään pyyntöinteraktion mukaan liittämällä interaktion nimeen **alaviiva** ja teksti **Response,** elementin namespace on "urn:hl7-org:v3" kuten muillakin elementeillä**,** esim PRSC\_IN030101FI01\_Response
* Vastaussanoman juurielementtiin lisätään attribuutti **intermediateLayer** joka saa arvon **true** ja on aina mukana sanomassa. Attribuutti määritellään seuraavasti: <attribute name="intermediateLayer" type="boolean" use="required" fixed="true"/>
* Vaihtoehtoiset vastausinteraktiot luetellaan WSDL tiedostossa <choice> rakenteen sisällä.

Esim. SOAP rakenne

<SOAP-ENV:Envelope>

<SOAP-ENV:Header>

.....

<SOAP-ENV:Header>

<SOAP-ENV:Body>

<PRSC\_IN030101FI01\_Response intermediateLayer="true" xmlns="urn:hl7-org:v3">

<PRSC\_IN030102FI01>

.....

</PRSC\_IN030102FI01>

</PRSC\_IN030101FI01\_Response>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

WSDL-esimerkki dokumentista ”Ajanvarausrajapinnat - Tekninen liittymämäärittely”:

Interaktion PRSC\_IN030101FI01 vastaussanoman juurielementti on PRSC\_IN030101FI01\_Response. Vaihtoehtoiset vastausinteraktiot juurielementin sisällä ovat PRSC\_IN030102FI01, PRSC\_IN030103FI01 ja MCCI\_IN000002UV01

WSDL kuvaus

<types>

---

<element name="PRSC\_IN030101FI01\_Response">

<complexType>

<choice>

<element ref="hl7:MCCI\_IN000002UV01"/>

<element ref="hl7:PRSC\_IN030102FI01"/>

<element ref="hl7:PRSC\_IN030103FI01"/>

</choice>

<attribute name="intermediateLayer" type="boolean" use="required" fixed="true"/>

</complexType>

</element>

</schema>

</types>

---

<message name="PRSC\_IN030101FI01\_Message\_Response">

<documentation>PRSC\_IN030101FI01 / New Appointment Slot Request Response</documentation>

<part name="Body" element="hl7:PRSC\_IN030101FI01\_Response"/>

</message>

---

<portType name="PRSC\_AR030101\_PortType">

<operation name="PRSC\_AR030101\_PRSC\_IN030101FI01">

<input message="hl7:PRSC\_IN030101FI01\_Message" wsa:Action="urn:hl7-org:v3:PRSC\_IN030101FI01"/>

<output message="hl7:PRSC\_IN030101FI01\_Message\_Response" wsa:Action="urn:hl7-org:v3:PRSC\_IN030101FI01\_Response"/>

---

</portType>

#### Yhteensopivuus

WS-I Basic profile 1.0 listaa parhaat periaatteet, joita noudatetaan käytettäessä SOAP 1.1:tä ja WSDL 1.1:tä. HL7 WSP noudattaa WS-I basic profilea.

**HL7-WSP300**: SOAP sanomien ja WSDL-kuvauksien tulee noudattaa WS-I Basic Profilea 1.0.

Kyseinen profiili ohjeistaa SOAP 1.1:n käytön ja käytämme Suomessakin SOAP 1.1:ta. Erot eivät ole kovin suuret teknisesti SOAP 1.2:een verrattuna. HL7:n WSDL-soveltamistapa ohjeistaa myös SOAP 1.2:n käytön, sillä WSDL 1.1:llä voidaan kuvata SOAP 1.2:n käyttö. WSDL 2.0:aa ei kuitenkaan käytetä.

WS-I basic profilesta on ilmestynyt myös versio 1.1, mutta erot edelliseen versioon eivät ole kovin suuria. Versiossa 1.1 on korjattu edellisen version virheitä ja ohjeistettu liitteiden siirto.

WS-I profiilin noudattamisessa on perusongelman vaatimuksen R2710 kohdalla. Kyseinen määritys vaatii, että wire signaturen, eli käytännössä SOAP-sanoman method wrapperin pitää olla yksikäsitteinen. HL7:ssa asiaa on tulkittu niin, että wire signature on payloadin juurielementin lisäksi SOAP action. Ongelma tulee esille kuittaussanomien kohdalla, mutta ei ole edes tekninen ongelma.

Kuten edellä esitetystä huomaamme, HL7 web services implementaatio on toteutettu suoraviivaisesti ja yksinkertaisesti. Sovellustason interaktiot muodostavat operaatiot ja operaatioista on yksi yhteen mäppäys vastaavaan siirtokuvaukseen.

**HL7-WSP400**: Tämän perusprofiilimäärityksen versio on ”http://www.hl7.org/v3/ws/2005/09/basic”

Kyseistä klausuulia (HL7-WSP400) voidaan käyttää määrityksen mukaisuus – dokumenteissa (conformance statements).

Tässä dokumentissa perusprofiilimääritystä on siis paikallistettu syyskuun 2005 äänestyspaketin mukaisesti. Sama määritys on May 2007 ballot-paketissa. Normative version 2006 sisältää vain basic profilen, joten käytämme ballot-pakettien määrityksiä.

### Osoitteistusprofiliili (HL7WSAP)

Perusprofiilissa määritellään jo wsa:action käyttö.

Osoitteistuksen tarvitsemat SOAP-headerit riippuvat perusliikennetyypistä (MEP).

**HL7 WSA100**: HL7 WS sanomien tulee noudattaa SOAP-liikentessä WS-addressing standardia.

**HL7 WSA200:** Tämän vaatimusmäärittelyn versio on ”http://www.hl7.org/v3/ws/2005/09/addressing”

HL7 Finland määrittää, että tätä profiilia käytetään myös Suomessa.

HL7 Addressing profile perustuu W3C-profiliin http:///www.w3.org/2005/08/addressing, joka on vielä kehitysvaiheessa. Edellä mainittua namespacea siis käytetään SOAP-headerissä. Yleisimmin käytetty namespace prefix on wsa.

HL7 Finland tarkennukset

HL7:n osoitteistusprofiili on melko ”kevyt”, siinä lähinnä vaaditaan WS addressing määritysten käyttöä.

Suomessa on CDA-dokumenttien siirrossa (avoimet adapterit) käytetty useita SOAP-headereitä, jotka on pääosin lainattu ebXML:stä. WS-addressing spesifikaatiota ei tuossa vaiheessa ollut vielä olemassa. Nyt kun V3 messaging otetaan käyttöön, kannattaa siirtyä kv. yhteensopivuuteen. Jatkossa myös CDA dokumentit voidaan Suomessakin kuljettaa V3 sanomien sisällä. CDA R1/R2 liikennöintitapa on ollut siirtymäkauden ratkaisu socket/MLLP-liikenteestä SOAP:iin.

V3 sanomaliikenne on SOAP 1.1:tä, koska perusprofiilit niin määrittävät. V3 messaging SOAP-header on melko kevyt, sillä sanomien hallintaan liittyvät seikat löytyvät sanoma- ja kontrollikehyksistä.

:

Erot open CDA-siirron ja V3-siirron välillä (SOAP-headerissä):

|  |  |
| --- | --- |
| open CDA SOAP-header | V3 messaging vastine |
| AckRequested | Interaktio määrittelee, halutaanko kuittaus |
| Acknowledgment | Reply- ja kuittaustyyppisissä sanomissa käytetään SOAP-header elementtiä RelatesTo ja siirtokehyksessä acknowledgement- targetMessage – id |
| HL7FIAcknowledgment | Nämä asiat ilmoitetaan siirtokehyksessä acknowledgementdetailissa |
| HL7FIBodyCount | Tarvittaessa voi käyttää BATCH-headeriä |
| MessageHeader-From-PartyId | **<From><address>** (saattaa poistua WSA-standardista) |
| MessageHeader-From-Role | Interaktio määrittelee |
| MessageHeader-To-PartyId | **<To>-**elementti |
| MessageHeader-To-Role | Interaktio määrittelee |
| MessageHeader-CPAId | Määritysdokumentti ilmoitetaan siirtokehyksen elementissä profileId. |
| MessageHeader-ConversationId | <**RelatesTo**>-elementti  Tarvittaessa voi käyttää reliable messagingiä |
| MessageHeader-Service | Interaktio määrittelee |
| MessageHeader-Action | **<Action>-**elementti, jossa on interaktion tunnus |
| MessageHeader-MessageData\_MessageId | **MessageId** jasiirtokehyksen id. |
| MessageHeader-MessaData-TimeStamp | Siirtokehyksessä creationTime. |

Sanomaliikenteen osapuolet voivat kuvata osoitteistuksen käytön EPR-elementillä (endpoint reference). Tästä elementistä voi olla viittaus WSDL:ään. EPR:n rakennetta ja siirtoa ei tässä kuvata, se löytyy WS-addressing standardista.

Yleensä osapuolten sähköiset osoitteet ovat URL-muodossa. Suomessa osapuolet tunnistetaan OID-koodilla, joten osapuolten osoitteet ilmoitetaan muodossa urn:oid (IETF RFC3001) , esim. urn:oid:1.2.246.10.6280613.18.1.

Suomessa käytettävät WS-addressing abstraktit käsitteet mappautuvat SOAP-headereiksi seuraavalla tavalla:

WS-Addressing elementit SOAP-headerissä WS-siirrossa, normatiivinen HL7 Finland

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Kuvaus** | **Tietotyyppi** | **SOAP-Header** |
| [destination] | vastaanottajan osoite | xs:anyURI | <wsa:To> </wsa:To> |
| [source endpoint] | lähettäjän osoite | EndpointReferenceType | <wsa:From>  <wsa:Address>  </wsa:Address>  </wsa:From> |
| [reply endpoint] | vastauksen osoite | EndpointReferenceType | <wsa:ReplyTo>  <wsa:Address>  </wsa:Address>  </wsa:ReplyTo> |
| [fault endpoint] | virheen osoite | EndpointReferenceType | <wsa:FaultTo>  <wsa:Address>  </wsa:Address>  </wsa:FaultTo> |
| [action] | Toiminto | xs:anyURI | <wsa:Action> |
| [message id] | viestin id | xs:anyURI | <wsa:MessageId> |
| [relationship] | kaksi arvoa, jotka ilmoittavat viestin suhteen toiseen viestiin | xs:anyURI, xs:anyURI  (ensimmäinen elementin arvona [id], toinen RelationshipType attribuutin arvona, oletus http:www.w3.org /2005/08/addressing/reply) | <wsa:RelatesTo RelationshipType= ”xs:anyURI”> |
| [reference parameters] | prioriteetti |  | ks. alla |

WS-Addressing elementtien käyttö HL7 V3 sanomaliikenteessä Kanta-palvelujen kanssa kommunikoitaessa

Kanta-palvelujen oma erillinen ohjeistusdokumentti (Osapuolitiedot Kanta-sanomaliikenteessä), joka on saatavissa www.kanta.fi –portaalin arkkitehtuuriosiosta. Määrittely ei enää edellytä HL7 V3 sanomaliikenteessä WS-Addressing elementtien käyttöä, poikkeuksena Sähköisen lääkemääräyksen uusimispyynnön vältys.

Esimerkki vastausinteraktion SOAP-headerin elementeistä (laboratoriovastaus):

xmlns:wsa=”http://www.w3.org/2005/08/addressing”

<wsa:To>urn:oid:1.2.246.10.7846981.18.2004.6543</wsa:To>

<wsa:From>

----

<wsa:Address>urn:oid:1.2.246.10.6280613.18.2004.225</wsa:Address>

</wsa:From>

<wsa:Fault>

<wsa:Address>mailto:timo.tarhonen@tto.fi</wsa:Address>

</wsa:Fault>

<wsa:Action>urn:hl7-org:v3:POLB\_IN224200</wsa:Action>

<wsa:MessageId>urn:oid:1.2.246.10.6280613.18.2004.225.2005.21221</wsa:MessageId>

<wsa:RelatesTo>urn:oid:1.2.246.10.7846981.18.2004.654.2005.10</wsa:RelatesTo>

Pyynnössä on ollut samantapaiset elementit, paitsi RelatesTo ja lisäksi:

<wsa:ReplyTo>

----

<wsa:Address>urn:oid:1.2.246.10.7846981.18.2004.6543</wsa:Address>

</wsa:ReplyTo>

<wsa:MessageId>urn:oid:1.2.246.10.7846981.18.2004.654.2005.10</wsa:MessageId>

Seuraavassa taulukossa on kuvattu perusliikennetapausten (MEP) edellyttämät SOAP-headerit. HL7FI-merkinnät ovat HL7-Finland lisäsuositukset käytettävistä elementeistä. Tosin From-elementti saattaa poistua WS-addressingin seuraavissa versioissa. Perusliikennetapaukset ovat normaali yksisuuntainen sanoma sekä kysely/vastaus.

P=pakollinen, HL7FI=HL7 Finland suositus

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **One-Way** | **Request** | **Response** |
| [destination] | P | P | P |
| [action] | P | P | P |
| [source endpoint] | HL7FI | HL7FI | HL7FI |
| [reply endpoint] |  |  |  |
| [fault endpoint] |  |  |  |
| [message id] | HL7FI | P | HL7FI |
| [relationship] |  |  | P |
| [reference parameters] | HL7FI | HL7FI | HL7FI |

Reply endpoint –header suositellaan jätettäväksi synkronisessa liikenteessä pois käytöstä.

### Tietoturvaprofiilli (HL7WSSP)

Suomessa perusprofiili ja osoitteistusprofiili ovat oleellisia yhteensopivan liikenteen toteuttamisen kannalta. HL7:ssa määritellään myös muita lisäpalveluja tarjoavia profiileja, joita osapuolet voivat ottaa käyttöön keskinäisellä sopimuksella.

Normaalisti Suomessa on käytetty turvalista siirtokanavaa 2WAY SSL. Se toteuttaa point-to-point – turvallisen siirron. Jatkossakin se on käypä tapa, koska liikennettä aiotaan ohjata viestinvälityskeskuksien kautta.

Jatkossa voidaan tarvita myös tietoturvallista sanomanvälitystä epäluotettavan verkon yli.

Web services tietoturvan yleiset periaatteet määritellään WS-I Basic Security Profile 1.0:ssa ja HL7-määritykset ovat yhteensopivia sen kanssa.

Perusmekanismit määritellään standardissa Web Services Security: SOAP message security 1.0, joka tunnetaan nimellä WS-Security. Tämä standardi tuo ws-liikenteseen mukaan mm. eheyden, muuttamattomuuden, kiistämättömyyden ja tunnistamisen. Standardissa määritellään erilaisten security token:ien käyttö.

Muut standardit rakentuvat WS-Securityn päälle. Näitä ovat mm.

* WS-Trust, jonka avulla voidaan välittää security tokeneita
* WS-SecureConversation, jonka avulla voidaan luoda päästä päähän tietoturvallinen yhteys (myös epäluotettavan siirtokanavan yli)
* WS-Policy, joka määrittelee mekanismin tietoturvavaatimuksien kuvaamiselle
* WS-SecurePolicy, jonka avulla määritellään tietoturvavaatimukset

HL7 International ohjeistaa näiden määritysten käyttöä. Tiukimmillaan profiilissa määritellään, että SOAP body on allekirjoitettava ja allekirjoitus ja SOAP-Body on salattava.

Tämän profiilin käyttöä ei tarkemmin tässä vaiheessa ohjeisteta eikä se ole HL7 Finlandin perusvaatimus. Osapuolet voivat kuitenkin käyttää tätä profiilia kahdenkeskisellä sopimuksella. Joka tapauksessa on hyvä tietää, että kv. HL7 on määritellyt myös tietoturvallisen siirron toteuttamista.

### Luotettava sanomanvälityksen profiili (HL7RMWSP)

WS-RM – Reliable messaging standardi on protokolla, joka takaa luotettavan siirron siirtoprotokollatasolla. Protokolla perustuu SOAP-tason headereihin ja sillä taataan sanoman perillemenosta varmistuminen, duplikaattiesto ja sanomien saapuminen perille lähetysjärjestyksessä.

Tämän protokollan käyttöä ei tarkemmin tässä vaiheessa ohjeisteta eikä se ole HL7 Finlandin perusvaatimus. Osapuolet voivat kuitenkin käyttää tätä protokollaa kahdenkeskisellä sopimuksella. RM-protokolla tekee myös osittain tarpeettomaksi vanhan sequence number- protokollan, tosin sequence number -protokolla sijaitsee sovellustasolla eikä siirtoprotokollatasolla.

## Määrityksistä käytettävät versiot

|  |  |  |
| --- | --- | --- |
| Määritys | Versio | huom |
| WS-I basic profile (WSIBSP) | 1.0 | 1.1 on olemassa, erot eivät ole suuria |
| WSDL | 1.1 | WSI BSP 1.0 määrää. WSDL 2.0 on olemassa. |
| SOAP | 1.1 | WSI BSP 1.0 määrää SOAP 1.1:n. |
| XML | 1.0 | WSDL 1.1 määrää |
| HL7-määritykset | V3 2006 standardi ja may 2007 äänestyspaketti | . |

HL7-määrityksissä on yleensä määritykset myös SOAP 1.2:lle. Työkalutuki on parempi 1.1:lle kuin 1.2:lle. HL7-sanomien siirto ei myöskään tarvitse 1.2:n uusia piirteitä. SOAP 1.1:n ja 1.2:n eroista on dokumentti Open CDA 2003 paketissa HL7:n dokumenttiarkistossa.

## Http-siirto

SOAP 1.1:ssa content-typen on oltava "text/xml".

SOAP 1.2:ssa content typen on oltava "application/soap+xml". HTTP Headerissa SOAPAction elementti on poistettu ja korvattu "application/soap+xml" media tyypin valinnaisella parametrilla "action".

POST-elementissä olevan palvelun osoite on implementointikohtainen.

Ohessa HTTP Header elementit  HL7 v3 laboratoriovastaussanomassa POLB\_IN224200, joka lähetetään osoitteeseen <http://palvelut.sairaala.fi/POLB_AR034000>. Vastaanottavan järjestelmän Application Role on POLB\_AR034000 (Result Receiver).

SOAP 1.1:

POST /POLB\_AR034000 HTTP/1.1  
Host: palvelut.sairaala.fi  
Content-Type: text/xml; charset=utf-8;  
Content-Length: nnnn   
SOAPAction: "urn:hl7-org:v3:POLB\_IN224200"  
  
<soap:Envelope xmlns:soap="...">  
  <soap:Header/>  
  <soap:Body>  
    <POLB\_IN224200 xmlns="urn:hl7-org:v3" ...>  
       ...  
  </soap:Body>  
</soap:Envelope>

SOAP 1.2:

POST /POLB\_AR034000 HTTP/1.1  
Host: palvelut.sairaala.fi  
Content-Type: application/soap+xml;   
charset=utf-8; action="urn:hl7-org:v3:POLB\_IN224200"  
Content-Length: nnnn   
  
<soap:Envelope xmlns:soap="...">  
  <soap:Header/>  
  <soap:Body>  
    <POLB\_IN224200 xmlns="urn:hl7-org:v3" ...>  
       ...  
  </soap:Body>  
</soap:Envelope>

# Semanttisen sisällön implementointi

## HL7-mallinnuksen perusperiaatteet

HL7 V3 standardissa kaikki mallinnustyö tehdään yleisen HDF- mallin mukaisesti. HDF on lyhenne sanoista HL7 Development Framework. Mallinnustyön pohjana on yleinen RIM-malli (Reference Information Model).

RIM-mallissa perusrakennuspalikat ovat:

act tapahtuma/havainto

act-relationship edellisten linkitys

participation osallistuminen

role rooli

entity fyysinen objekti

relationship link roolien välinen suhde



Kuva 29 - RIM:in perusrakenne

RIM:issä on näistä perusobjekteista tehty periyttämällä yleiskäyttöisiä objekteja. RIM perustuu UML-standardiin. RIM on pyritty rakentamaan siten, että sitä pohjana käyttäen voidaan kuvata maailmankaikkeuden kaikki tapahtumat.

RIM-mallista tehdään sovellusalueen yleiskuvaus D-MIM (domain information model) siten, että RIM:in rakenneobjekteja monistetaan kulkemalla act-relationship:in kautta ja uusille acteille ja niihin liittyville muille objekteille annetaan uudet nimet. Puhutaan kloonauksesta. D-MIM rakennetaan MS Vision avulla.

Sovellusten välisten interaktioiden toteuttamista varten tehdään R-MIM-malleja (refined message information model). Loogisesti R-MIM periytyy D-MIM:istä. Teknisesti R-MIM rakennetaan samalla tavalla kuin D-MIM RIM-objekteista kloonaamalla. RIM ikään kuin tarjoaa erimuotoisia (vain tietyllä tavalla yhdistettävissä olevia) legopalikoita, joita saadaan monistaa ja värittää uudelleen ja sen jälkeen yhdistää halutulla tavalla. R-MIM rakennetaan myös Visiolla.

R-MIM:istä tehdään hierarkisia (siirtotekniikasta riippumattomia) sanomakuvauksia, joista puolestaan generoidaan XML-schemoja.

Kaikilla sovellusalueilla toimitaan samoilla periaatteilla (HDF) ja samoilla työkaluilla.



Kuva 30 – Kullakin sovellusalueella sanomat kehitetään samalla tavalla

**HDF**



Sanomakuvauksia tehdään siis R-MIM:ien avulla. Kun lopullinen schema generoidaan työkaluilla, saadaan mukaan yleiset tietotyyppimäärittelyt ja sanasto. Scheman generointiin on oma säännöstönsä, jonka XML ITS toteuttaa.

## Vocabularyn periaatteet

Vocabularyyn liittyy käsite **Realm** (vanha nimi Application Contex), Realm voi olla universaali (UV) ja jokainen HL7 affiliate muodostaa oman Realminsa, Suomen Realm on ”FI”.

### Vocabulary domain

V3 malleissa jokaiselle koodatulle attribuutille on määritelty Vocabulary Domain. Vocabulary Domain on abstrakti käsite, joka on aina universaali eikä sitä ole kiinnitetty mihinkään koodistoon. Vocabulary Domain käsitteen avulla attribuutissa käytettävät arvot (Value Set) voidaan määrittää vasta myöhemmin sanomakehityksen aikana ja eri Realmeissa voidaan käyttää eri koodistoja.

### Value Set

Implementaation yhteydessä attribuutin abstrakti Vocabulary Domain liitetään tietyssä Realmissa koodistoon ja näin saatuja attribuutissa käytettäviä koodiarvoja kutsutaan nimellä Value Set. Value Setillä ei ole usein nimeä eikä OID koodia, ja jos OID on määritelty se on tarkoitettu vain conformance dokumenteissa (esim. implementaatio-oppaissa) käytettäväksi. Sanomassa attribuutin kentissä käytetään aina Value Setin sisältävän koodiston nimeä ja OID koodia.

Yksi Value Set sisältää aina vain yhden koodiston arvoja, mutta Value Set voi sisältää toisia Value Settejä ja tätä kautta arvoja useista koodistoista.

Value Set arvot voidaan määrittää luettelemalla sallitut arvot (Extensional Value Set Definition) tai ilmoittamalla koodisto ja mahdolliset rajoitukset (Intensional Value Set Definition).

Joillakin HL7:n määrittelemillä Value Seteillä on nimi ja OID, joiden root on 2.16.840.1.113883.11. Näitä ei siis tule käyttää sanomainstansseissa, joissa käytetään arvot sisältävän koodiston nimeä ja OID koodia.

Periaatteessa yhdessä Realmissa tietty Vocabulary Domain on kiinnitetty yhteen Value Setiin, mutta käytännössä eri implementaatio-oppaissa voidaan määrittää samalle Vocabulary Domainille eri Value Setit (ja siten eri koodistot). Paikallinen koodisto muodostaa oman Value Setin ja se voidaan V3:n käsitteistössä joko sitoa tiettyyn Vocabulary Domainiin tai liittää osaksi toista Value Settiä.

### Code system

V3:ssa käytetään mahdollisimman paljon olemassa olevia ulkopuolisia koodistoja ja omia koodistoja kehitetään vain jos tarvittavia koodistoja ei ole. HL7 V3 koodistot ovat standardissa ja ballot paketeissa kohdassa Foundation/Vocabulary

Suomen kansallisten koodistojen koodistopalvelimen osoite on http://sty.stakes.fi/FI/koodistopalvelu/koodisto.htm.

Jonkin verran sekavuutta aiheuttaa HL7:n omaksuma käytäntö nimetä Vocabulary Domain, Value Set ja Code system samalla nimellä esim ActCode on sekä Vocabulary Domain, Value Set (OID 2.16.840.1.113883.11.13953) että koodisto (OID 2.16.840.1.113883.5.4). Sanomassa käytetään aina koodiston OIDia.

### OID

Sanomissa käytettävien HL7 koodistojen OID tunnukset löytää HL7 OID Registrystä osoitteesta <http://hl7.org/oid/index.cfm>.

HL7 OID tunnukset on ryhmitelty seuraavasti:

2.16.840.1.113883.1 HL7 Internal Objects (HMDs, rMIMs, etc)

2.16.840.1.113883.2 HL7 Organization Bodies - TCs, SIGs, affiliates etc

2.16.840.1.113883.4 HL7 specified root for commonly used external identifier systems

2.16.840.1.113883.5 HL7 maintained code systems

2.16.840.1.113883.6 External coding systems registered in HL7 with an HL7 OID

2.16.840.1.113883.7 HL7 published documents (standards, implementation guides, etc

2.16.840.1.113883.11 HL7 Value Sets

2.16.840.1.113883.12 HL7 v2 tables (Suomen paikallistetut V2 taulut 1.2.246.537.5.”taulun numero”)

2.16.840.1.113883.19 Examples Root; not to be used in production

OID tunnusten käyttö Suomessa noudattaa seuraavia ohjeita:

JUHTA JHS 159

<http://www.jhs-suositukset.fi/intermin/hankkeet/jhs/home.nsf/files/JHS159/$file/JHS159.pdf>

Stakes: ”ISO OID yksilöintitunnuksen käytön kansalliset periaatteet sosiaali- ja terveysalalla” suora linkki tällä hetkellä:

http://sty.stakes.fi/NR/rdonlyres/C0432B2A-7AB6-4D4E-AFE2-3927962AE423/9180/OIDopasversio1020070214.pdf

OID yksilöintitunnusten avulla yksilöidään kaikki objektit ainutlaatuisesti. Tunnisteesta ei tule päätellä objektin semanttisia ominaisuuksia. OID luokitushaarojen ohjeistus on tarkoitettu pelkästään auttamaan OID tunnisteiden generoinnissa.

## Lähtökohta

Paikallistamistyössä voidaan keskittyä varsinaiseen sovellusalueeseen, sillä esim. siirrossa tarvittavat wrapperit ja siirtotekniikkaan liittyvät määritykset on kuvattu toisaalla tässä dokumentissa. Muita yleisiä tukidokumentteja pyritään myös aikaansaamaan pikaisesti.

## Perusperiaate

HL7 V3 malli on laaja ja siinä useat osat ovat sidoksissa toisiinsa. HL7 V3 on edelleen kehitysvaiheessa, vaikka sitä onkin kehitetty jo monta vuotta. Semanttiselle sisällölle tehdään kauaskantoista mallinnusta, joka ei ole sidottu varsinaisiin siirtostandardeihin. Koska kehitystyö jatkuu, niin voi olla työlästä paikallistaa (suomentaa) koko tietyn sovellusalueen (domainin) sisältö. *Tämän vuoksi implementointioppaissa pitäisi pystyä kertomaan suomeksi sovellusalueen yleinen juoni, keskeiset komponentit sekä tarkennukset kansainväliseen standardiin sekä paikallistamistyössä mahdolliset tehdyt sääntöjen mukaiset maakohtaiset lisäykset.* Tavoitteena on, että implementointioppaan päivittäminen olisi helppoa, vaikka kansainvälinen standardi muuttuukin.

## HL7-domainit

Tällä hetkellä (2007) julkaisutietokannassa on seuraavat sovellusalueet:

Health and Clinical Management Domains

|  |  |  |  |
| --- | --- | --- | --- |
| Subsection | Lyhenne | Domain | Lyhenne |
| Operations | PO |  |  |
|  |  | Bloodbank | POBB |
|  |  | Clinical Document Architecture | POCD |
|  |  | Clinical Genomics | POCG |
|  |  | Laboratory | POLB |
|  |  | Pharmacy | PORX |
|  |  | Medication | POME |
|  |  | Informative Public Health | PORI |
|  |  | Public Health Reporting | PORR |
|  |  | Regulated Studies | PORT |
|  |  | Specimen Domain | POSP |
|  |  | Therapeutic Devices | POTD |
| Records | RC |  |  |
|  |  | Medical Records | RCMR |
|  |  |  |  |
| Reasoning | RE |  |  |
|  |  | Care Provision | REPC |

Administrative Management Domains

|  |  |  |  |
| --- | --- | --- | --- |
| Subsection | Lyhenne | Domain | Lyhenne |
| Practice | PR |  |  |
|  |  | Patient Administration | PRPA |
|  |  | Scheduling | PRSC |
|  |  | Personnel Management | PRPM |
| Financial | FI |  |  |
|  |  | Claims & Reimbursement | FICR |
|  |  | Accounting and Billing | FIAB |

Infrastructure Management

|  |  |  |  |
| --- | --- | --- | --- |
| Subsection | Lyhenne | Domain | Lyhenne |
| Message Control | MC |  |  |
|  |  | Message Control Infrastructure | MCCI |
|  |  | Message Act Infrastructure | MCAI |
| Master File | MF |  |  |
|  |  | Master File Management Infrastructure | MFMI |
| Query | QU |  |  |
|  |  | Query Infrastructure | QUQI |
| Common Message Elements | CO |  |  |
|  |  | Common Message Elements | COCT |
|  |  | Shared Messages | COMT |

## Artifaktien nimeäminen

HL7 määritykset sisältävät useita erilaisia objekteja, joita kutsutaan artifakteiksi.

Artifaktityyppien lyhenteet

|  |  |
| --- | --- |
| Artifact | Code |
| Application Role | AR |
| D-MIM (Domain Information Model) | DM |
| HMD (Hierarchical Message Descriptor) | HD |
| Interaction | IN |
| Message Type | MT |
| R-MIM (Refined Message Information Model) | RM |
| Storyboard | ST |
| Storyboard Narrative | SN |
| Trigger Event | TE |

Yksittäiset artifaktit nimetään seuraavalla kaavalla:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Subsection | Domain | “\_” | Artifact | 6 numeroa | UV(=Realm) | Versionumero |

Esim.

PRPA\_AR000001UV01

PR = Subsection: Practice

PA = Domain: Patient Administration

AR = Artifact: Application Role

000001 = 6 numeroinen yksikäsitteinen tunnus

UV = Realm (UV = Universal on virallisen kansainvälisen standardin mukaisissa määrityksissä)

01 = versio

Versionumero kasvaa aina kun määritys on hyväksytty äänestyskierroksella. Realm-koodilla erotetaan eri maiden paikallistetut versiot. Esim. Suomen paikallistetuissa määrityksissä realm on FI.

Äänestyksessä olevissa hyväksymättömissä versioissa ei ole realm koodia ja numeroa vaan ne ovat muodossa esim. PRPA\_IN111111 tai PRPA\_IN111111UV ja vasta hyväksymisen jälkeen käytetään muotoa PRPA111111UV01.

Nimeämiskäytännön tulkinnan osaaminen helpottaa huomattavasti standardiin perehtymistä.

## Implementointioppaan muoto

*Implementointioppaan liitteeksi pitää ottaa se julkaisutietokannan (publishing database) versio, johon paikallistamistyö perustuu.* Joissakin maissa paikallistamistyö on tehty julkaisutietokannan perusteella (UK NHS). Kansainvälisissä keskusteluissa on kuitenkin tullut ilmi, että julkaisutietokannan käyttö implementointioppaan pohjana on työlästä, vaatii paljon resursseja ja osaamista. Jotkut ovat jopa sitä mieltä, että julkaisutietokanta ei ole edes tarkoitettu implementointioppaiden tekoon. HL7:n piirissä ollaankin kehittämässä työkaluja implementointioppaiden tekemisen tueksi. HL7 julkaisutietokannassa informaatio on linkitetty web-tekniikan avulla. *Suomessa työkaluna kannattaa siis edelleen käyttää word-tekstinkäsittelyä ja taulukkolaskentaa (excel)* (kunnes saamme käyttöön parempia työkaluja). Implementointioppaan käyttäjällä on siis samaan aikaan auki word-dokumentti (ja excel) ja toisessa ikkunassa hän selaa julkaisutietokannasta vastaavan sovellusalueen dokumenttia.

## Sovellusalueen yleiskuvaus

HL7 julkaisutietokannassa on sovellusalueen yleiskuvaus ja siihen liittyvä D-MIM (domain information model). *Yleiskuvaus suomennetaan ja peruskäsitteistä ja terminologiasta tehdään luettelo, jossa nähdään suomenkielisen ja englanninkielisen käsitteen vastaavuus. D-MIM malli liitetään dokumenttiin.*

## Interaktiopohjainen lähestymistapa

HL7-standardissa varsinaiseen ongelmaan pureudutaan käyttötapauksien (use case) kautta, jotka on kuvattu käyttötapauskuvauksissa (storyboard). Käyttötapauksien pohjalta on määritelty käyttötapauksen integraatiotarpeiden toteuttamiseksi tarvittavat interaktiot.

*Paikallistamisessa suomennetaan tarvittavat käyttötapaukset.*

Interaktioon liittyy aina sen käynnistävä liipaisin (trigger). Interaktion yhteydessä on määritelty myös lähettäjän ja vastaanottajan roolit (application roles). Interaktion sanomamalli on kuvattu graafisesti R-MIM-mallissa (refined message information model). R-MIM malli (tehty Visiolla) on serialisoitu siirtoa varten hierarkiseksi sanomakuvaukseksi HMD (hierarchical message description). Hierarkinen sanomakuvaus ei sisällä vielä mitään siirtotekniikkaan liittyvää.



Kuva 31 – sanomanvälityksen ja mallinnuksen suhde

Interaktion yhteydessä määritellään, mitä vastaanottajan pitää tehdä (receiver resposibilities) sanoman saatuaan (jos mitään), luettelemalla interaktio/interaktiot, joilla ko. interaktioon on vastattava. Vastuut määritellään niille interaktioille, joille odotetaan sovellustason vastausta ( = sovellustason kuittausta). Interaktion määrittelyn receiver responsibilities osio määrittelee mahdolliset vastausinteraktiot ja triggerit.

Jokaisen interaktion receiver responsibilities osioon kuuluu interaktio MCCI\_IN000002UV01 eli Accept Ack vaikka sitä ei ole erikseen mainittu. Eli vastaanottajan on aina kyettävä vastaamaan vastaanottokuittauksella jos lähettäjä sitä siirtokehyksessä pyytää.

Esimerkki pharmacy domainista:

Prescription Topic

2.1 Storyboards

2.2 Application Roles

2.3 Trigger Events

2.4 Refined Message Information Models

2.5 Hierarchical Message Descriptions

2.6 Interactions

Sovellusalueen yleismallin (D-MIM) tutkimisen, yleiskuvauksen ja käyttötapauskuvauksien tutkimisen jälkeen seuraavaksi kannattaa tutkia interaktioiden luetteloa. *Luettelosta valitaan suomenkieliseen dokumenttiin Suomessa käytettävät interaktiot.*

HL7-standardissa kunkin interaktion kohdalla on määritelty seuraavat seikat:

* liipaisin (trigger event)
* siirtokehys (transmission wrapper)
* kontrollikehys (control act wapper)
* sanomatyyppi (message type)
* lähettäjän ja vastaanottajan roolit
* receiver responsibilities (vastausinteraktiot)

Sanomatyypin kautta päästään:

* R-MIM-kuvaukseen
* Hierarkiseen sanomakuvaukseen

Hierarkisen sanomakuvauksen kautta on määritys ja linkki mm. seuraaviin asioihin:

* Luokkamäärittely
* RIM-attribuutti
* tietotyyppi
* sanasto

Useampi kuin yksi sanomakuvaus voi käyttää samaa hierarkista sanomakuvausta.

Kun sovellusalueen yleinen juoni on selvillä, interaktioluettelo on siis erinomainen entry interaktiossa tarvittaviin teknisiin määrityksiin.

Esimerkki laboratoriopyynnön interaktiosta:

Order Fulfillment Request with Receiver Responsibilities (POLB\_IN111100)

Description

|  |  |
| --- | --- |
| Structured Name | Interactions With Receiver Responsibilities Order Activate Fulfillment Request |
| Trigger Event | Order Activate  POLB\_TE001100 |
| Transmission Wrapper | Send Message Payload  MCCI\_MT000100UV01 |
| Control Act Wrapper | Trigger Event Control Act  MCAI\_MT700201UV01 |
| Message Type | Placer Order POLB\_MT001000 |
| Sending and Receiving Roles |  |
| Sender | Order Placer POLB\_AR010000 |
| Receiver | Order Fulfiller POLB\_AR020000 |

Receiver Responsibilities

|  |  |  |
| --- | --- | --- |
| Reason | Trigger Event | Interaction |
| The receiver is confirming receipt of the order fulfillment request | POLB\_TE001002 | POLB\_IN121000 |
| The receiver is rejecting the order fulfillment request | POLB\_TE001003 | POLB\_IN121001 |
| The receiver is responding with a promise to fulfill the order | POLB\_TE002100 | POLB\_IN122103 |
| The receiver is responding with a completed result that fulfills the order | POLB\_TE004200 | POLB\_IN124202 |

Interaktioon liittyviä teknisiä yksityiskohtia ei tarvitse määritellä, sillä ne löytyvät julkaisutietokannasta. *R-MIM kuvat voidaan halutessa liittää paikallistamisdokumenttiin.*

Semanttisten implementointiin liittyvien yksityiskohtien määrittely pitäisi periaatteessa tehdä siirron aikaista tasoa ylemmällä tasolla, mikä tarkoittaa hierarkisen sanomakuvauksen käyttöä tai R-MIM:iä. Hierarkinen sanomakuvaus on kuitenkin taso, jossa suositellaan dokumentoitavaksi sovellustoimittajienkin määrityksen mukaisuudet (conformance statements), joten se sopii myös kansallisen implementointioppaan lähtökohdaksi.

Hierarkisessa sanomakuvauksessa on linkki siihen RIM-luokkaan, josta se on peritty. Itse asiassa HMD:stä on linkki kaikkiin ”tietokentissä” käytettyihin määrityksiin. Erityisen hyödyllinen on tietokenttien kohdalla oleva suora linkki sanastoon (Vocabulary), josta nähdään käytettävissä olevat arvot.

*Hierarkisen sanomakuvauksen taulukosta otetaan elementin numero ja nimi, tehdään uusi taulukko, jossa kunkin elementin kohdalla kerrotaan implementointiohje.* Taulukko toteutetaan teknisesti word-dokumentin kapppaleina otsikon ”hierarkinen sanomakuvaus” alla. Jatkossa mietitään myös excelin käyttöä.. *Valituissa kohdissa esitetään esimerkkiarvo yleisellä tasolla sekä XML-koodattuna. Luokitukset ja koodistot tulee dokumentoida ja uusista luokituksista tehdään liitedokumentti (vietäväksi STAKES:in koodistopalvelimelle).*

*Interaktion sanomamäärityksen rakenteesta tehdään graafinen yleiskuvaus word-dokumenttiiin jollain XML-työkalulla (esim. XML SPY). Lisäksi kuvataan muut tarvittavat yksityiskohtaisemmat rakenteet, jotka määrityksen tekijä kokee keskeisiksi.*

*Ainakin yhdestä interaktiosta tehdään täydellinen siirtovalmis esimerkki.*

## Muita ohjeita

Implementointioppaalle annetaan OID-koodi. Implementointioppaan mukaisissa sanomissa implementointioppaan OID-koodi sijoitetaan siirtokehyksen elementtiin profileId. Implementointiopas määrittelee myös käytettävän HL7 V3 version. Kyseinen tieto sijoitetaan sanomissa siirtokehyksen kenttään versionCode.

# Jatkokehitys

Tämä dokumentti on kehitetty vuosien 2005,2006 ja 2007 aikana open CDA projektissa.

Dokumentin pääpaino sanomien kuorikerroksien kuvaamisessa ja Web services siirron ohjeistamisessa.

Dokumenttia tarkennetaan myös tulevaisuudessa implementointiprojektien myötä.

# Esimerkki

Liitetiedostossa v3esimerkki.xml on esimerkki siirtovalmiista aineistoista. Kyseessä on SerAPI-projektin ajanvarausrajapintojen määrittelypaketin esimerkkisanoma PRSC\_IN030102FI01 interaktio SOAP sanomana. Esimerkin tarkoituksena on havainnollistaa Suomeen paikallistettua kontrollikehyksen, siirtokehyksen sekä SOAP-headerin sisältöä sekä SOAP-rakennetta.

Todellisessa siirrossa näkyvät http-headerit on kuvattu kappaleessa 3.4.