

HL7 Development Framework - taustaselvitys

Versio 1.5

	SerAPI-projekti	
	Yhteyshenkilö	Mika Tuomainen (mika.tuomainen@uku.fi)
	Dokumentin tila	Työversio
	Päiväys	28.3.2006

Sisällysluettelo

1	HL7 Development Framework	4
2	HL7 Versio 3.0.....	5
2.1	Yleistä versiosta 3.....	5
2.2	V3:n kehittämisfilosofia.....	5
2.3	V3 kehitysmenetelmän avainkäsitteet	6
2.3.1	UML.....	6
2.3.2	RIM-malli.....	6
2.3.3	MDF/HDF	7
3	HL7 Message Development Framework (MDF)	8
4	HDF vs. MDF	12
4.1	Soveltamisalan kattavuus.....	12
4.2	Yhdenmukaistettu UML:n kanssa	12
4.3	Menettelyt ylläpitoon/versiointiin	12
5	HDF Methodology Specification: kehitysprosessi.....	13
5.1	Projektin aloittaminen (Project Initiation)	15
5.2	Vaatimusten dokumentointi (Requirements Documentation).....	16
5.2.1	Toimintaprosessin dokumentointi (Document Business Process).....	17
5.2.2	Prosessin etenemisen dokumentointi (Capture Process Flow).....	19
5.2.3	Rakenteen dokumentointi (Capture Structure).....	19
5.2.4	Toimintasääntöjen dokumentointi (Capture Business Rules)	20
5.2.5	Sovellusalueen analyysimallin harmonisointi (Harmonize the Domain Analysis Model with HL7 Reference Models).....	22
5.3	Mallinnus (Specification Modeling).....	24
5.3.1	Staattisten tietonäkymien suunnittelu (Build Design Models of Static Information Views)	24
5.3.2	Toiminnallisten näkymien suunnittelu (Construct Design Models of Behavioral Views)	24
5.3.3	Uudelleenkäytettävien mallien määrittely (Define Reusable Design Model Components).....	24
5.3.4	Vuorovaikutuksen suunnittelu (Construct Design Models of Collaboration and Interaction)	24
5.3.5	Suunnittelussa syntyneiden mallien harmonisointi (Harmonize Design Models with HL7 Reference Models)	24
5.4	Määrittelyn dokumentointi (Specification Documentation)	24
5.5	Määrittelyn hyväksyminen (Specification Approval)	24
5.6	Määrittelyn julkistaminen (Specification Publication).....	24
5.7	Toteutuksen ja käyttöönoton profilointi (Implementation Profiling)	24
6	HL7 UML Profile & Metamodel	24
7	HDF Developers Guides.....	24
7.1	HDF Developer's Guide Vol 1 - Construction of a Design Level Static Model	24
7.2	HDF Developer's Guide Vol 2 - Information Harmonization	24

Versiohistoria

Versio:	Pvm:	Laatijat:	Selitys:
Versio 0.1	15.12.2004	Mika Tuomainen	Pohjadokumentti erilaisista kalvosarjoista ja HL7 uutiskirjeestä.
Versio 0.2	29.12.2004	Mika Tuomainen	Runko valmis, materiaalia HDF-projektin dokumenteista. Mukana V3 taustaa.
Versio 1	14.1.2005	Mika Tuomainen	Perusasiat, joista dokumenttia voidaan lähteä tarvittaessa laajentamaan.
Versio 1.5	27.3.2006	MT, Juha Mykkänen	Laajennukset tammikuun 2006 Ballot-paketin pohjalta mallinnusta ja suunnittelua koskeviin osiin (vaiheet 2 ja 3) luvussa 5.

Esipuhe

Tämä työ liittyy SerAPI-hankkeeseen (Palveluarkkitehtuuri ja Web-sovelluspalvelut Terveystieteiden Ohjelmistotuotannossa ja -integraatiossa), jossa tutkitaan ja kehitetään web-sovelluspalvelujen ja palvelupohjaisen arkkitehtuurin hyödyntämistä terveydenhuollon tietojärjestelmätarpeisiin ja sovellusintegraatioon ja uusiin sekä olemassa oleviin ohjelmistotuotteisiin. Hanketta rahoittavat Tekes (päätös nro 40437/04) sekä joukko yrityksiä ja sairaanhoitopiirejä.

1 HL7 Development Framework

HL7 Development Framework (HDF) on kehys/metelmä HL7:n (Health Level Seven) käyttämien prosessien, käytäntöjen ja tuotosten mallinnukseen ja hallintaan. HL7 käyttää menetelmää terveydenhuollon tietojärjestelmien yhteentoimivuutta tukevien standardien kehittämiseen.

HDF:n tavoitteena on käyttää mallipohjaista lähestymistapaa kaikkiin HL7-määrittelyihin. Lisäksi HDF:ssä yhdenmukaistetaan HL7:ssä käytettyjä mallinnuskäytäntöjä (modelling practices) standardiin Unified Model Language (UML) merkintä- ja kuvauskieleen.

HDF on Message Development Framework:n (MDF) korvaaja ja laajennus. MDF:ää on käytetty mallipohjaisena menetelmänä HL7 Version 3.0 (V3) viestimäärittelyjen tuottamisessa. MDF-metelmää on kuvattu tarkemmin luvussa 3.. HDF eroaa MDF:stä läheisemmällä UML:n käytöllä, sillä HDF:n alla oleva metamalli on linjattu UML:n metamallin kanssa. Lisäksi HDF:n tavoitteena on käyttää mallipohjaista menetelmää kaikkiin HL7 teknisiin määrittelyihin, ei ainoastaan sanomamäärittelyihin.

HDF:n kehitystä johtaa HDF-projekti. Projektin tarkoituksena on tutkia, analysoida, suunnitella ja dokumentoida menetelmät, käytännöt ja artefaktit, jotka liittyvät HL7 standardien määrittelyyn. HDF-projekti alkoi tammikuussa 2002 ja sillä on tällä hetkellä useita tuotoksia draft-tasolla. Näitä projektin päätuotoksia ovat:

- HDF Methodology Specification (viimeisin versio 01/2006)
- HL7 UML Profile & Metamodel (05/2004)
- HDF Developer's Guide Vol 1 - Construction of a Design Level Static Model (5/2004)
- HDF Developer's Guide Vol 2 - Information Harmonization (9/2004).

Koska HDF-projekti on vielä kesken, HL7 Version 3.0 standardimäärittelyjen kehittäminen ei edellytä vielä HDF-menetelmän käyttämistä.

Projektin vetovastuu on HL7:n Modeling & Methodology teknisellä komitealla. HL7 Technical Steering Committee vastaa projektin johtamisesta, valvomisesta ja kiistanalaisten kysymysten ratkaisemisesta. HDF-projektia ovat sponsoroineet Health Level Seven, Inc. sekä osittaisella rahoituksella Veterans Health Administration (VHA). Projektin hallinnoinnista on vastannut Shakir Consulting.

2 HL7 Versio 3.0

2.1 Yleistä versiosta 3

HL7 teki vuonna 1992 päätöksen muuttaa standardiensa kehitysmenetelmää. Uusi menetelmä sai nimen HL7 Version 3.0 (V3). V3:sta alettiin kehittää, kun HL7 huomasi, että V2:sen suosiosta huolimatta tarvittiin järeämpi menetelmä aidosti yhteensopivien järjestelmien tukemiseksi. V3 on mallipohjainen menetelmä, joka perustuu olio-ohjelmointisuuntautuneeseen sovelluskehitykseen. Siinä käytetään moderneja analysointi tekniikoita niin järjestelmän rakentamisessa kuin viestien suunnittelussa. Keväällä 1997 kaikki HL7:n tekniset komiteat alkoivat käyttää V3-menetelmää.

HL7 V3 on pisimmälle kehitetty HL7 standardi. Se sisältää enemmän laukaisevia tapahtumia (trigger events) ja sanomamuotoja kuin aikaisemmat versiot. Siinä yhdistyvät HL7 RIM-käsitelmä (Reference Information Model), määritellyt sanastot ja terminologiat (vocabulary specifications), mallipohjainen analyysi (model-driven analysis) sekä suunnitteluprosessit. Yksi V3:n päätavoitteista on olla standardi, joka on tarkka ja testattavissa oleva ja joka tarjoaa toimittajille mahdollisuuden todistaa, että heidän toteutuksensa on määritysten mukainen (certification of conformance).

HL7 V3:n tavoitteena on myös sisältää kehitysmenetelmä, jossa kehitetään konsensukseen perustuvia standardeja terveydenhuollon tietojärjestelmien väliseen yhteistoiminnallisuuteen. Menetelmään liittyy edelleen kehittyvä määrittelyprosessi. Uusin versio V3-standardien kehitysmenetelmästä on HDF (Health Level Seven Development Framework).

2.2 V3:n kehittämisenfilosofia

V3-standardien kehitysmenetelmä perustuu joukkoon periaatteita, jotka tarjoavat yleisen filosofian kaikille V3-standardien kehittäjille:

- Kansainvälisyys: V3:n sisältää mekanismin, jolla voidaan tukea paikallisia muunnoksia HL7 sanomista.
- Tuki perinnejärjestelmille.
- Löyhästi kytketyt järjestelmät: V3 ei standardoi järjestelmien toiminnallisuutta, se ainoastaan kuvailee vaatimukset toteutuksille.
- V2 toiminnallisuuden tukeminen.
- Vastaavuuksien määrittelyt määrittelyjen mukaisuuden todentamiseksi: Vastaavuusväitteet perustuvat sovellusten rooleihin.
- Potilastietojen luottamuksellisuus.
- Autentikoidut ja valtuutet palvelut.
- Turvallisuus, yksityisyys, kiistämättömyys ja eheys.

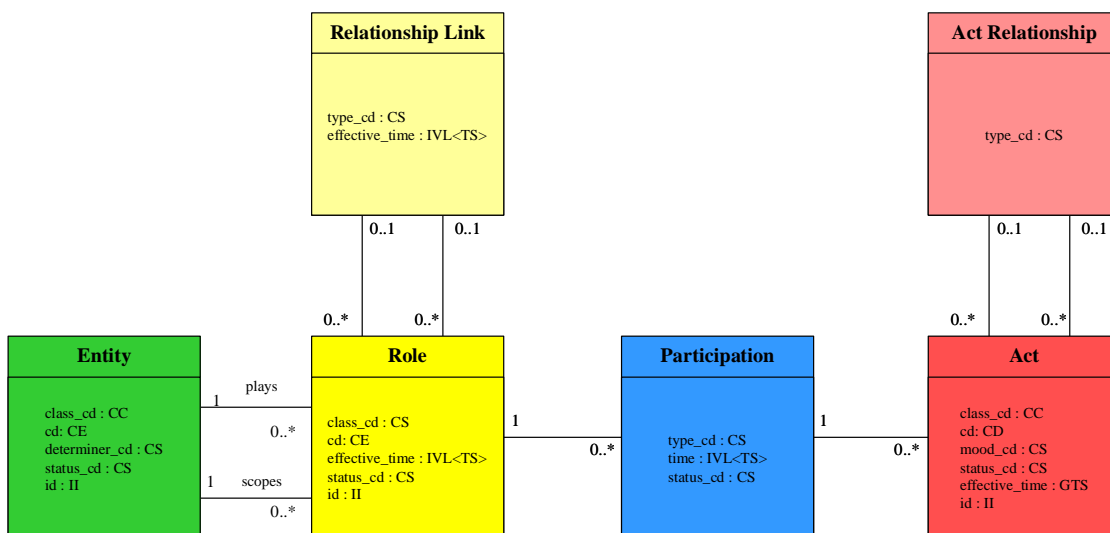
2.3 V3 kehitysmenetelmän avainkäsitteet

2.3.1 UML

HL7 V3 kehitysmenetelmä määrittelee oliopohjaisen mallintamisen avulla tiettyyn terveydenhuollon toimeen liittyvät tärkeimmät tiedot ja semantiikat. Menetelmässä käytetään **UML:n (Unified Modelling Language)** keskeisiä elementtejä.

2.3.2 RIM-malli

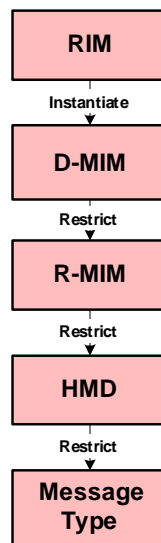
HL7 Reference Information Model (RIM) on yhteinen jaettu tietomalli. Se on tärkeä osatekijä V3-menetelmässä. RIM-mallista johdetaan kaikki muut tietomallit ja -rakenteet. RIM on ilmaistu käyttämällä UML-notaatiota (kuva 1). RIM-malli muodostuu luokista, jotka ovat terveydenhuollon toimialaan liittyvien subjektien tai muiden toimialaan liittyvien asioiden abstraktioita. RIM-mallin luokista voidaan tarkentaa tarkempia luokkia, joita tarvitaan tietyssä sanomassa tai sanomaryhmässä. RIM-mallin kaikki luokat pohjautuvat kuvassa 1 esitettyihin kuuteen perusluokkaan (core classes).



Kuva 1. RIM-mallin perusluokat (Hinchley)

Viestien määrittelyssä (kuva 2) RIM-mallista johdetaan **D-MIM** (Domaine Message Information Model), joka on tietyn sovellusalueen tarkennettu viestimalli. D-MIM:stä määritellään **R-MIM** (Refined Message Informantion Model), joka on viestin tarkennettu tietomalli. Kun R-MIM on määriteltä, voidaan siitä määritellä itse viestejä. Viestit tarkennetaan vielä **HMD**:llä, hierarkkisella viestin kuvauksella (Hierarchical Message Description).

HL7 V3 kehitysmenetelmä määrittelee säännöt em. tietomallien johtamiseen RIM-mallista. Säännöt vaativat, että kaikki RIM:stä johdetut tietomallit, pitää olla jäljitettävissä takaisin RIM-malliin ja että niiden semantiikka ja niihin liittyvät toimitasäännöt eivät ole ristiriidassa RIM:ssä määritelyjen kanssa. RIM toimii näin kaikkien HL7 V3 standardien tietosisältöjen perimmäisenä lähteenä.



Kuva 2. Viestien määrittely (Shakir 2003a).

2.3.3 MDF/HDF

HL7 V3:ssa käytettiin aluksi HL7 Message Development Framework (MDF) -menetelmää, joka on mallipohjainen prosessi HL7 V3:sen viestien kehittämistä/rakentamista varten. MDF tullaan korvaamaan HDF:llä, jossa mallipohjaisuus tulee pysymään avainkäsitteenä. MDF:ää käydään läpi tarkemmin seuraavassa luvussa.

3 HL7 Message Development Framework (MDF)

HL7 Message Development Framework:ssä (MDF) kuvataan mallipohjainen prosessi HL7 V3:sen viestien kehittämistä/rakentamista varten. MDF julkaistiin V3:n ensimmäisissä versioissa. MDF:ää ei kuitenkaan enää ylläpidetä eikä sen käyttämistä enää suositella. Tämän korvaajaksi on luotu HL7 V3 Guide for messaging-opas HL7 Working Group-jäsenten käyttöön. Opas ei ole yhtä laaja sisältöltään kuin MDF mutta se käsittelee MDF-menetelmään kohdistuneet uudistukset. Ajan mittaan HL7 Development Framework (HDF) tulee korvaamaan MDF:n. HDF on kuitenkin vielä draft-tasolla. HDF:stä tarkemmin luvussa 4.

Seuraavaksi käydään kuitenkin MDF-menetelmän perusasioita läpi sillä HL7 V3 Guide sisältää samat elementit kuin MDF ja HDF:n voidaan sanoa myös sisältävän MDF:n.

MDF-menetelmässä (kuva 3) tunnistetaan vaiheet, aktiviteetit ja mallit, joita käytetään HL7 V3 viestimäärittelyjen kehittämiseen. MDF menetelmä on muodollinen tapa ottaa huomioon tarvittavat vaatimukset sanomien määrittelyssä. Näin varmistetaan, että kaikki tiettyyn sanomaan tarvittavat tiedot saadaan tallennettua. MDF:ään on määritelty Reference Model Repository (RMR)-varasto, jonne viestien kehitysprosessin aikana tuotetut tuotokset varastoidaan ja jossa tuotoksia voidaan hallita.

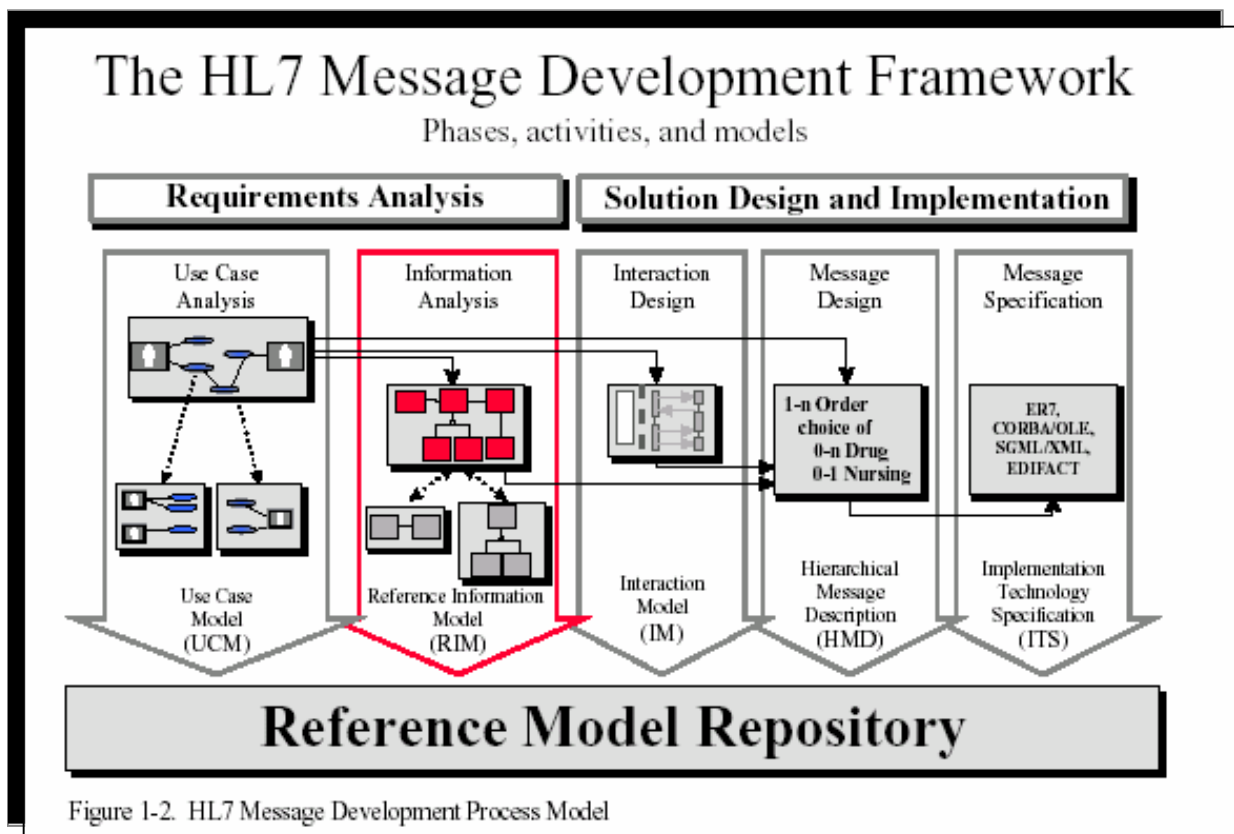
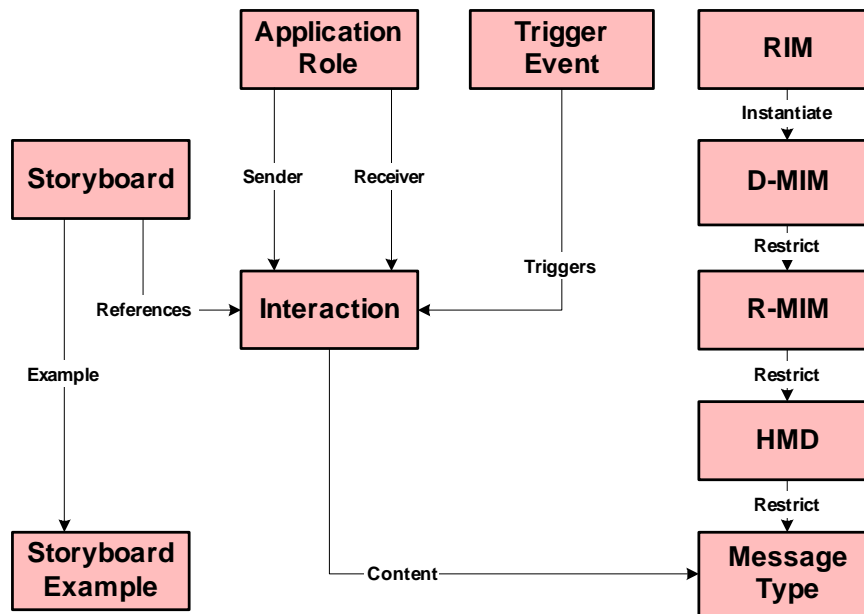


Figure 1-2. HL7 Message Development Process Model

Kuva 3. MDF-menetelmän yleiskuva (Shakir 2003a).

Kuvassa 4 on kuvattu yksinkertaistettu MDF-tietomalli.

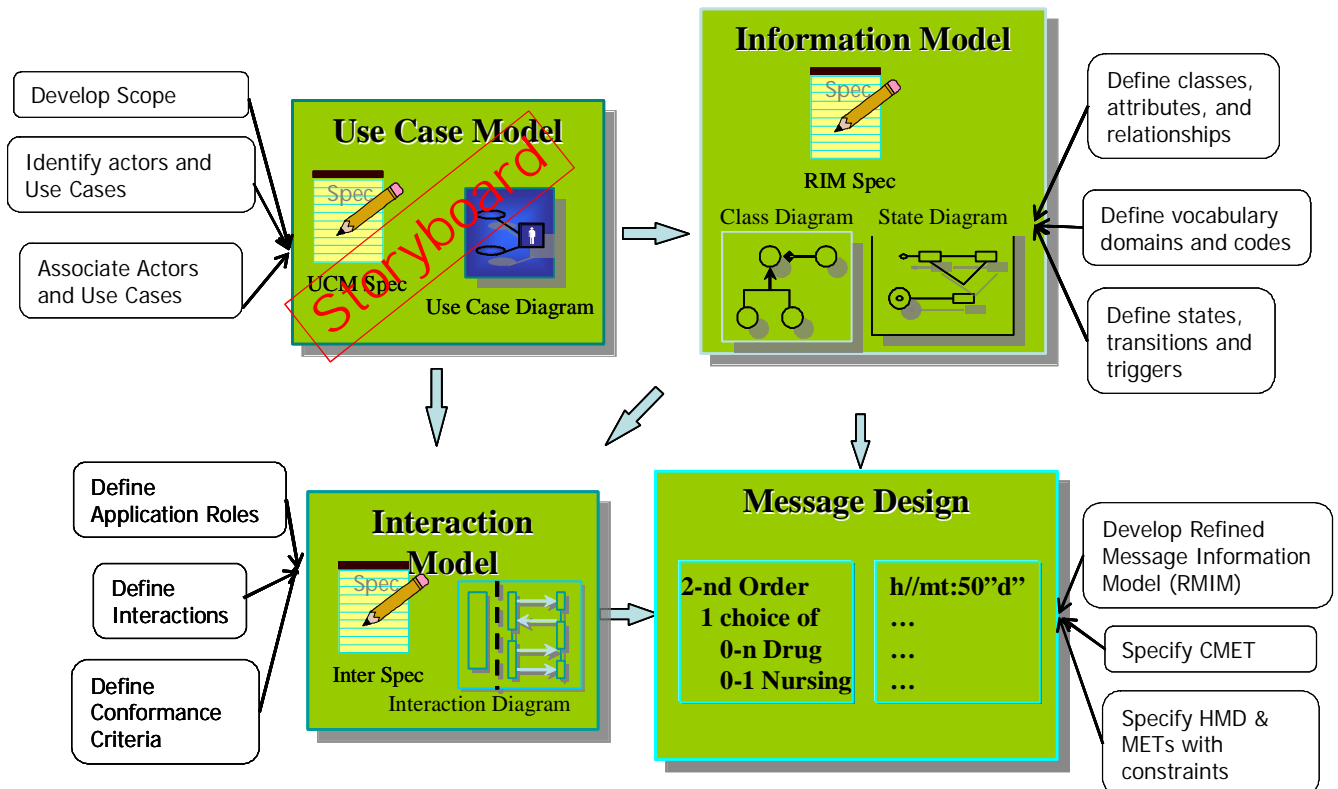


Kuva 4. Yksinkertaistettu MDF-tietomalli (Shakir 2003a).

Tietomallin osat:

- sovelluksen rooli (application role): Sovelluksen ominaisuus, joka tukee tiettyä V3 viestiä tai viestiryhmää. Roolissa kuvataan interaktion aikaiset viestiä lähettävän ja vastaanottavan sovelluksen vastuut.
- laukaiseva tapahtuma (trigger event): Tapahtuma terveydenhuollon prosessissa, joka tallennetaan tai tunnistetaan terveydenhuollon järjestelmässä ja joka luo tarpeen tietovirralle yhteen tai useampaan järjestelmään. Tapahtuma määrittelee syyn, miksi viesti lähetettiin
- storyboard ja storyboard-esimerkki: Lyhyt realistinen kuvaus tosielämän prosessista, jossa saatetaan tarvita viestiä. Storyboard:issa käytetään presens-aikamuotoa ja se seuraa prosessia alusta loppuun. Kuvaukset määritellään peräkkäisinä interaktioina tai käyttötapauksina.
- interaktio (interaction, vuorovaikutus, yhteistoiminta): Yksi yhdensuuntainen tietovirta, joka tukee skenaariossa esitettyjä vaatimuksia viestinnälle. Interaktio on vuorovaikutuskaavion (interaction model) yksi elementti, joka sisältää mm.:
 - lähettävän sovelluksen roolin
 - vastaanottavan sovelluksen roolin
 - määritellyn laukaisevan tapahtuman (trigger event)
 - määritellyn viestin tyyppin

Kuvassa 5 on kuvattu HL7 V3 viestien määrittelyn vaiheet (Heitmann ym. 2002). Em. yksinkertaistettu MDF-tietomalli linkittyy V3-menetelmän eri vaiheisiin.



Kuva 5. HL7 V3 viestien määrittelyn vaiheet (HL7 2002).

1. Storyboard - Kirjoitetaan yksinkertainen esimerkki sovellusalueelle, josta käy ilmi kuinka tietoa siirretään tai tulisi siirtää kyseessä olevassa tilanteessa. Tavoitteena on ymmärtää, mitkä toimijat osallistuvat toimintaan ja miten, mitkä ovat heidän tietotarpeensa ja milloin he tarvitsevat tietoa, sekä kuinka he hyödyntävät tietokonetta tehtävässä. Myös kuvaus on osa määrittelyn dokumentointia.
2. Sovellusroolit:
 - a. esimerkistä (storyboard) tutkitaan, missä järjestelmien välistä kommunikaatiota tarvitaan. Järjestelmiä käsitellään "tyypeinä" (sairaalatietojärjestelmä, laboratoriojärjestelmä jne.) ja suuret järjestelmät jaetaan pienempiin toiminnallisiin osiin (esim. ajanvaraus, laboratorio suuressa sairaalajärjestelmässä).
 - b. käytetään nuolia ilmaisemaan esimerkissä kuvattua tiedonvälitystä (esim. A kysyy B:ltä tuloksia, B vastaa).
 - c. kustakin tiedonvälityskohdasta päätetään, mikä on siirrettävän tiedon pääsisältö (esim. lähete, rtg-tutkimuksen tulokset, tilaus jne.).
 - d. näiden tietojen perusteella listataan sovellusten roolit.

3. Interaktiot:

- a. piirretään karkean tason yhteistoimintakaavio, jossa sovellusroolit ovat laatikoita ja tietovirrat suunnattuja nuolia niiden välillä.
- b. kukin nuoli kuvaa interaktiota, ja sillä on tunniste, laukaiseva tapahtuma (trigger event), ja siirtyvän viestin nimi tai yhteenveto siirrettävän viestin sisällöstä.

4. Laukaisevat tapahtumat (trigger events), sovellusroolit, viestityypit ja interaktiot lisätään tietokantaan (publication database), jossa säilytetään käytetyt viestit ja muut määrittelyt uudelleenkäyttöä varten.

5. Kullekin interaktiolle määritellään viestin sisältö. Yleistä sanastoa käyttäen listataan tietoelementit, jotka on siirrettävä (esim. lähete, jossa mukana potilaan henkilötiedot, lähetävä lääkäri, etäkonsultaatiopyyntö, laboratoriotutkimuspyyntö viimeisten 10 päivän aikana tehdyistä tutkimuksista jne.)

6. Järjestetään ja strukturoidaan kohdan 5 listat, tavoitteena dokumentoida interaktioiden ja tietojen riippuvuudet, tietoelementtien mahdollinen ryhmittely jne.

7. Ryhmittelyn perusteella päätetään, mitkä tietokokonaisuudet suunnitellaan itse (oma tiimi), ja mitkä saadaan valmiina aikaisemmista määrittelyistä tai muista ryhmistä (committee).

8. Itse suunniteltavat tietojoukot luokitellaan RIM- tai (R-MIM)-mallien mukaisiin perusluokkiin (core classes):

- Toiminnot (Acts),
- Toimintosuhteet (Act_relationships),
- Osallisuudet (Participations),
- Entiteetit (Entities),
- Roolit (Role),
- Linkit rooleihin (Role_link)

9. Käytetään välineiden (Visio) viestimalli (R-MIM) -notaatiota ja luodaan kustakin R-MIM-mallin osasta kaavio. Kaavioon otetaan mukaan kunkin osan todennäköiset tai tärkeimmät tietoelementit. Kullekin luokalle määritellään tunnistekoodi, ja toiminnoille myös tapahtumakoodi (mood code, esim. onko kyseessä suoritettu, aiottu, tavoite, riski, mahdollisuus tms.).

10. Linkitetään kaavion osiot yhteen R-MIM-mallin dokumentoimiseksi.

4 HDF vs. MDF

HDF on MDF:n korvaaja ja laajennus. HDF eroaa MDF:stä seuraavissa kohdissa:

- soveltamisalan kattavuus
- yhdenmukaistettu UML:n kanssa
- menettelyt ylläpitoon/versiointiin.

4.1 Soveltamisalan kattavuus

HDF dokumentoi kaikissa HL7 määrittelyissä tarvittavat prosessit, välineet, actorit, säännöt ja tuotokset. Se ei keskity ainoastaan sanomien määrittelyyn. Ensimmäiset versiot HDF menetelmämäärittelyistä (methodology specification) päivittivät sanomamäärittelyä sekä soveltuivat rakenteisten dokumenttien ja kontekstinhallinnan määrittelyyn (context management). HDF:n tavoitteena on lopulta käsittää kaikki HL7 standardimäärittelyt.

4.2 Yhdenmukaistettu UML:n kanssa

HDF-menetelmässä kehitetyt tuotosten (artefaktien) metamallit dokumentoidaan UML:n mukaisesti/mukaisesti. Aiemmin käytetty MDF poikkeaa jonkin verran UML -määrittelystä. Suurin osa poikkeuksista on vähäisiä mutta muutamia merkittäviäkin poikkeuksia löytyy.

HDF-projekti analysoi HL7 metamallin ja UML metamallin ristiriitaisuudet. Ristiriitojen ratkaiseminen johti HL7 UML Profile:n kehittämiseen. Tämä profiili hyödyntää UML:n laajennusmahdollisuuksia. Lopullinen UML-yhteensopiva HDF metamalli dokumentoidaan HDF Methodology Specification-määrittelyn tukidokumenttina.

Metamalli = Malli, jonka tarkoituksena on antaa kokonaisvaltainen kuva prosessista, järjestelmästä jne., erityisesti abstrahoidulla yksityiskohtaisilla malleilla, jotka liittyvät kyseiseen metamalliin.

4.3 Menettelyt ylläpitoon/versiointiin

HDF dokumentoi HL7 standardimäärittelyjen kehitysprosessit. Lisäksi se dokumentoi käytännöt ja menetelmät, jotka liittyvät prosessien kehittämiseen ja versionhallintaan. HDF Methodology Specification on "elävä dokumentti", jota voidaan muuttaa systemaattisella ja kurinalaisella tavalla itse menetelmän kehittyessä. HDF Methodology Specification -määrittelyyn on liitetty toimiviksi osoittautuneet tekniikat dokumenttien hallintaan, konfiguraation hallintaan ja versionhallintaan.

5 HDF Methodology Specification: kehitysprosessi

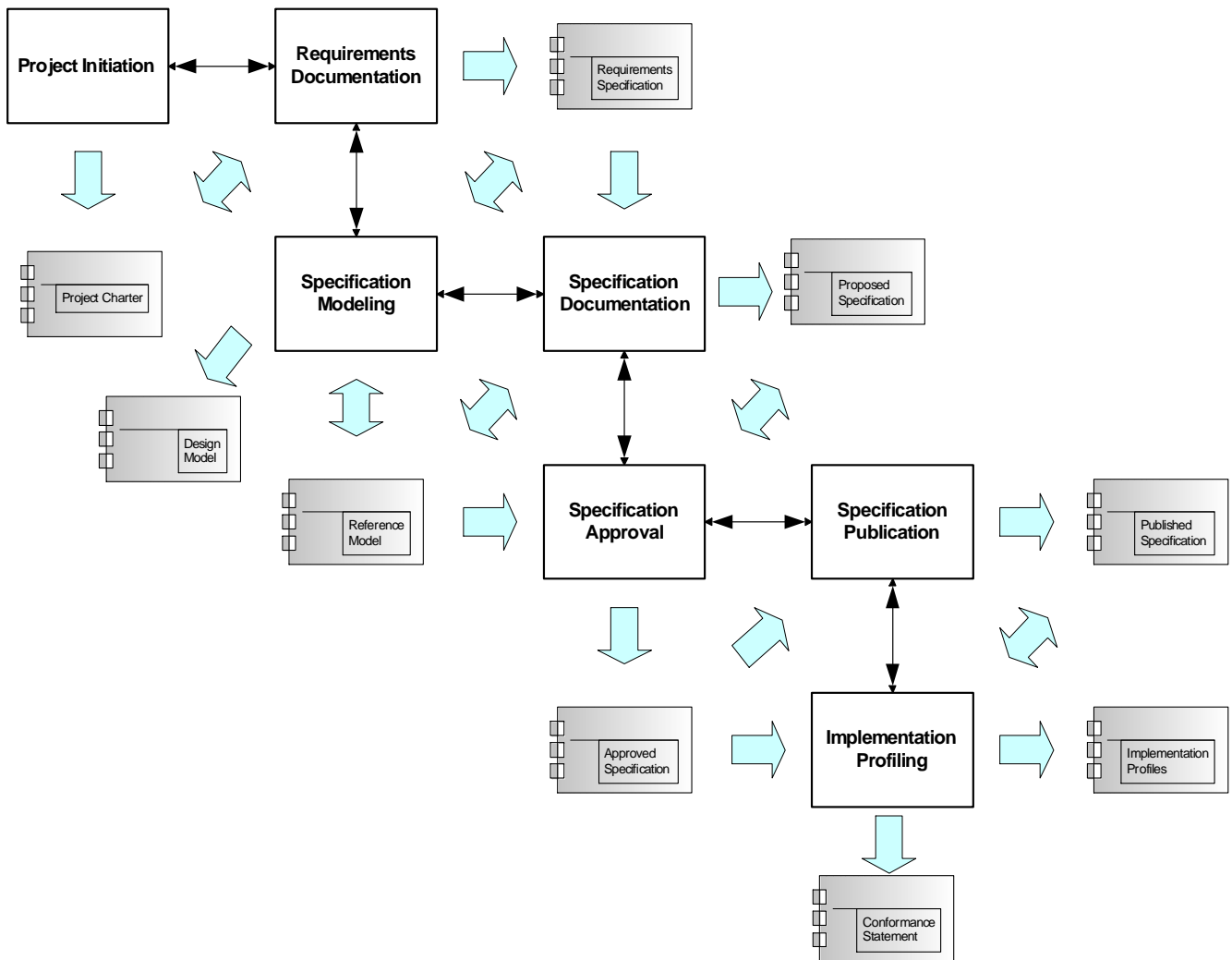
HL7 Development Framework (HDF) Methodology Specification -määrittelyn tavoitteena on tarjota yleiskuva ja konteksti määrittelyprosesseille sekä määrittelytyön tuotoksille käsittäen koko HDF-kehityksen elinkaaren. HDF-menetelmä esitetään abstraktina prosessina, joka on riippumaton tietystä tuotettavasta määrittelystä tai tietyistä työstettävistä kehitysprojektista. Menetelmän tavoitteena on esitellä prosessit, kuvata niissä tuotettavat tuotokset ja niiden riippuvuudet sekä toimia syötteenä tarkemmille tietojen määrittelyjen suunnitteluoppaille.

HDF-menetelmää kuvaavaa HDF Methodology Specification –dokumenttia on yksinkertaistettu jättämällä pois tarkoituksella kuvaukset HDF-metamallista ja määrittelytyössä käytettävistä välineistä. Niitä varten ovat omat tukidokumentit HDF UML Profile, josta löytyy HDF metamalli sekä HDF Developer's Guide Vol 1, jossa kuvataan HDF menetelmää käyttäen apuvälineenä Microsoft Visio -työkalua sekä HDF Developer's Guide Vol 2, jossa kuvataan tietomallien harmonisointia. HDF Methodology Specification -dokumentissa ei ole myöskään määrittelyä tarkasti ja yksityiskohtaisesti HDF-menetelmällä tuotettavien tuotosten muotoa.

HDF Methodology Specification –dokumentissa määritellään yleinen HDF-kehitysprosessi, joka on sovellettavissa kaikkiin HL7 standardien määrittelyyn. HDF kehitysprosessi sisältää seuraavat seitsemän vaihetta (kuva 6):

1. Projektin aloittaminen (Project initiation)
2. Vaatimusten dokumentointi (Requirements Documentation)
3. Määrittelyn mallinnus (Specification Modelling)
4. Määrittelyn dokumentointi (Specification Documentation)
5. Määrittelyn hyväksyminen (Specification Approval)
6. Määrittelyn julkistaminen (Specification Publication)
7. Toteutuksen ja käyttöönoton profilointi (Implementation Profiling).

Mallinnus liittyy tarkemmin näihin kohdissa "2 Vaatimusten dokumentointi" ja "3 Määrittelyn mallinnus". Nämä molemmat liittyvät otsikon "Modeling, Analysis, and Harmonization" alle. Myös kohdassa "7 Toteutuksen ja käyttöönoton profilointi" käytetään osittain mallinnusta. Kohdassa Määrittelyn dokumentointi voidaan muokata vaiheissa 2 ja 3 tuotettuja malleja, mutta niitä ei mallinneta enää uudelleen.



Kuva 6. HDF-kehitysprosessi (Shakir 2003a).

Jokainen vaihe on kuvattu seuraavissa kappaleissa lyhyesti. Vaikka toiminnot on kuvattu lineaarisella tyylillä, ne eivät ole riippuvaisia tietyistä järjestyksestä. Eri toiminnot ovat kuitenkin implisiittisesti riippuvaisia toimintojen tuotoksista. Yhden toiminnon tuotos toimii toisen toiminnon syöteenä. Prosessia on kuitenkin tarkoitus toteuttaa iteratiivisella tavalla, eli myöhempien vaiheiden palaute otetaan huomioon uudelleen aiemmissa vaiheissa, jos se on tarpeellista.

HDF Methodology Specification -dokumentti kuvaa jokaisen vaiheen toiminnot, osallistujat (vastuut), mahdollisesti käytettävät apuvälineet ja tuotokset. Näitä on listattu dokumentissa seuraavansiin taulukkoihin (taulukko 1)

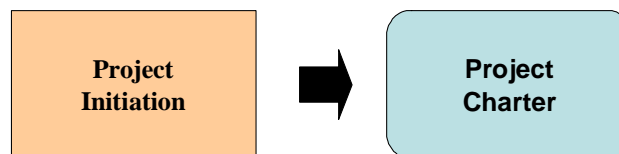
Taulukko 1 (HL7 2006).

INPUTS REQUIRED:	
PROCESS:	
TOOLS:	
RESPONSIBILITY:	
OUTPUTS:	
NOTES:	
WHEN:	

Seuraavissa luvuissa käydään tarkemmin läpi HDF-kehitysprosessin vaiheita, ja erityisesti mallinukseen ja ratkaisumäärittelyyn liittyviä vaiheita on käyty läpi tarkalla tasolla.

5.1 Projektin aloittaminen (Project Initiation)

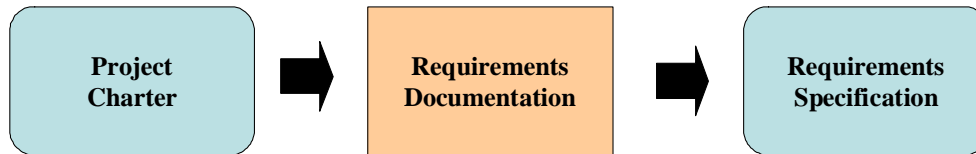
Projektin aloittamisvaiheessa projekti määritellään, tuotetaan projektisuunnitelma ja saavutetaan projektin hyväksyntä. Projektin aloittamisvaiheen päätuotos on projektin perustamiskirja (project charter).



1. Määritellään projektin vaikutusalue, tavoitteet ja suunnitellut tuotokset.
2. Tunnistetaan projektiin osallistujat (stakeholder, participant) ja tarvittavat resurssit.
3. Dokumentoidaan projektin olettamukset, rajoitteet ja riskit.
4. Valmistellaan alustava projektisuunnitelma ja dokumentoidaan riippuvuudet projektin sisällä
5. Hyväksytetään projekti ja aloitetaan se.

5.2 Vaatimusten dokumentointi (Requirements Documentation)

Vaatimusten dokumentoinnissa määritellään sovellusalueen ongelma, tuotetaan sovellusalueen malli ja harmonisoidaan tuotettu malli HL7:n viitemalleihin.



Vaatimusten dokumentoinnin päätuotos on vaatimusmäärittely, tarkemmin kuvattuna tuloksia ovat:

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

Vaatimusten dokumentaatioissa on seuraavat vaiheet:

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

HDF:n vaatimukseen liittyvissä osissa esiteltyjen tekniikoiden tavoitteena on mahdollistaa HL7-vaatimusten määrittely ja dokumentointi selkeästi sekä verifioitavalla ja testattavalla tavalla. Tavoitteena on saada rakennettua yksimielisyys kaikkien osapuolten (stakeholders) välille sekä kirjattua kaikki odotukset.

Tässä käytävien tekniikoiden avulla dokumentoidut toimintaprosessien dynaamiset, vastuisiin perustuvat interaktiot ovat muunnettavissa järjestelmien välisiksi interaktioiksi, sovellusroolien määrittelyiksi sekä sovellusroolien vastuiksi. Lisäksi dynaamisen käyttäytymisen mallintamistekniikoita voidaan käyttää selventämään tietyn sovellusalueen HL7-triggereitä sekä tilan muutoksia.

Lähestymistapa on tarkoitettu käytettäväksi iteratiivisesti ja inkrementaalisesti. Tarkoitus ei ole siis edetä vesiputousmallin tyyliin "kerätään kaikki vaatimukset ennen kuin edetään seuraavaan vaiheeseen". Vaatimusten hyväksyntä tehdään jäsenäänestyksen kautta (membership ballot).

Alla käydään tarkemmin läpi vaatimusten dokumentoinnin vaiheet.

5.2.1 Toimintaprosessin dokumentointi (Document Business Process)

Ensimmäinen askel vaatimusten keräämisessä on kehittää kuvaus tietystä terveydenhuollon sovellusalueen ongelmasta (business problem) tai -prosessista (process). Tämä tehdään storyboardin avulla. Storyboard on esimerkkikuvaus sovellusalueesta. Storyboard esittää yksityiskohtaisesti ajallisesti järjestyksessä sarjan toimia tai interaktioita, jotka koskettavat yhtä tai useampia osallistuvia entiteettejä (eli ihmisiä ja/tai järjestelmiä), ja joilla on jokin tietty merkitys (arvo) yhdelle tai useammalle osallistuvalla osapuolella. Storyboard vastaa kysymykseen, minkä vuoksi tietoa on tarve jakaa.

Jokainen HL7 määrittely tukee terveydenhuollon näkökulmasta määriteltyjä ja rajattuja toimintaprosesseja. Suurin osa näistä määrittelyistä on lisäksi keskittynyt mahdollistamaan tiedonsiirtoa ja/tai interaktioita kahden tai useamman terveydenhuollon järjestelmän välille.

INPUTS REQUIRED:	<ul style="list-style-type: none"> • Healthcare Business Process (Domain Expertise) • HL7 Project Charter (Business Strategy, Goals, Objectives, Risks, Dependencies, Constraints)
PROCESS:	<ul style="list-style-type: none"> • Documentation of a specific healthcare business process involves unambiguously describing both the structure and the behavior/function of the entities involved in the process. Documentation of the behavior/function of a particular business process is initially captured in a Storyboard. • HL7 Specification definition is a 'top-down' process that begins with: <ul style="list-style-type: none"> ○ a description of a specific healthcare business problem (or processes) that require(s) the exchange of data/information ○ and/or the interaction of healthcare information systems in order for the overarching healthcare business processes to be successfully executed • The data/information/interaction requirements are initially defined: <ul style="list-style-type: none"> ○ Between business systems, business entities, business workers, business participants, etc. ○ In an implementation-independent manner, i.e. a statement of the problem to be solved (i.e. a certain healthcare business process requires certain data to flow from one stakeholder to another) rather than a solution to be applied
TOOLS:	<ul style="list-style-type: none"> • Storyboard Example (See section 2.4, Work Products & Deliverables (WP&D). • Context Diagrams (yhteyskaavio) & Collaboration Diagrams (yhteistyökaavio)
RESPONSIBILITY:	<ul style="list-style-type: none"> • Domain Expert (A/R) • HL7 Facilitator (C)
OUTPUTS:	<ul style="list-style-type: none"> • Documented Dynamic Behavior and Static Structure
NOTES:	<ul style="list-style-type: none"> • The requirements for each HL7 Specification (or Specification component) must be traceable to a specific healthcare business process. This traceability relationship serves two important purposes: <ul style="list-style-type: none"> ○ Facilitation of validation of the Specification requirements domain experts unfamiliar with the design and implementation details of HL7 exchange artifacts (e.g. messages) to quickly determine whether a particular HL7 data/information artifact exist to support a given business-process-of-interest ○ Facilitation of complete and accurate Specification design and construction relative to the defined requirements (see the HDF chapter on Modeling, Analysis and Harmonization);
WHEN:	<ul style="list-style-type: none"> • The generation of the artifacts mentioned above is not done sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done

Yhteistyökaaviossa (collaboration diagram) keskitytään kuvaamaan, sitä miten yhteistyö hyödyntää olioiden välisiä kytkentöjä. Pohjana on oliokaavio. Yhteyksiin liitetään niiden kautta kulkevat viestit. Kuvausta käytetään samaan tarkoitukseen kuin sekvenssikaaviota. Kaaviolla voidaan varmistaa, että oliomalli tarjoaa tarvittavat kytkennät palvelujen saamiseksi (viestien välittämiseksi).

5.2.2 Prosessin etenemisen dokumentointi (Capture Process Flow)

Aktiviteettikaavioiden avulla kuvataan terveydenhuollon toimintaprosesseja (niiden aktiviteetteja ja etenemistä). Kun storyboardista on saatu nostettua esiin yksittäinen prosessi, se selvennetään ja käsitellään syvällisemmin aktiviteettikaavion avulla. Aktiviteettikaavioilla pyritään keskittymään sisäisen (järjestelmän tai järjestelmän osan) prosessin etenemisen kuvaamiseen.

INPUTS REQUIRED:	<ul style="list-style-type: none"> Documented Dynamic Behavior and Static Structure
PROCESS:	<ul style="list-style-type: none"> Clarify and expand the Storyboard into an Activity Diagram. The Activity Diagram will visualize the activities and flow of a healthcare business process.
TOOLS:	<ul style="list-style-type: none"> Activity Diagram (UML v1.4)
RESPONSIBILITY:	<ul style="list-style-type: none"> Domain Expert (A/R)
OUTPUTS:	<ul style="list-style-type: none"> Activity Diagram (See WP&D 2.4.3)
NOTES:	<ul style="list-style-type: none">
WHEN:	<ul style="list-style-type: none"> The generation of the artifacts mentioned above is not done sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done

5.2.3 Rakenteen dokumentointi (Capture Structure)

Tietyn toimintaprosessin rakenteen dokumentoimiseen käytetään sovellusalueen analyysimallia ja käsitteistöä. Sovellusalueen analyysimalli on UML-luokkakaavio. Luokkakaavio ei tässä ole toteutustasoinen luokkakaavio, vaan sen tarkoituksena on korostaa käsittelyssä olevan sovellusalueen "luokkien" tunnistamista ja niiden staattisia keskinäisiä suhteita. Nämä keskinäiset suhteet kuvataan UML:n assosiaatioiden (yhteydet), assosiaatioiden nimien ja kardinaliteettien (lukumääräsuhteet) avulla. Yleensä sovellusalueen analyysimallin "luokat" sisältävät vain muutaman (jos lainkaan) attribuutin ja luokan metodit (luokan "käyttäytyminen") mielletään usein "vastuiksi".

Käsitteistössä on kuvattu termit, joita tietyn sovellusalueen prosessiin kuuluu.

INPUTS REQUIRED:	<ul style="list-style-type: none"> • Documented Dynamic Behavior and Static Structure • Activity Diagram
PROCESS:	<ul style="list-style-type: none"> • Capture the vocabulary of the domain including the set of terms that domain experts use to describe the various components of the process (e.g. responsible entities, data elements, etc.) <ul style="list-style-type: none"> ○ Each new vocabulary term must be listed defined using 'definition' syntax and semantics as described in the HDF Style Guide (i.e. no self-referential definitions, included examples, etc.). ○ The collected set of domain-specific terms must then be refined into a Glossary and a Domain Analysis Model; the Domain Analysis Model pictorially documents the static syntactic and semantic relationships of importance in the healthcare business process including the responsible parties/entities and the various data elements/structures required by the process
TOOLS:	<ul style="list-style-type: none"> • UML Class Modeling Tools
RESPONSIBILITY:	<ul style="list-style-type: none"> • Domain Expert (A/R) • HL7 Modeling Facilitator (C)
OUTPUTS:	<ul style="list-style-type: none"> • Domain Analysis Model (See WP&D - 2.4.4) • Domain Glossary (See WP&D - 2.4.5)
NOTES:	<ul style="list-style-type: none"> • After several iterations, a comprehensive definition of the specific healthcare business process is developed with pictorial terms that a domain expert can understand, discuss, and refine. • The specific UML syntax required to refine an Activity Diagram or Domain Analysis model may need to be supplied by an HL7 Modeling Facilitator
WHEN:	<ul style="list-style-type: none"> • The generation of the artifacts mentioned above is not done sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done

5.2.4 Toimintasääntöjen dokumentointi (Capture Business Rules)

Toimintasääntöinä HDF:ssä käsitellään suhteet (Relationships), laukaisevat tapahtumat (Triggers) ja rajoitteet (constraints).

Seuraava askel vaatimusten dokumentointiprosessissa on huolellinen kuvaus tiedon rakenteesta, jota toimintaprosessia tukevassa tiedonvaihdossa käytetään. Tämä rakenne lisätään aktiviteettikaavioon käyttämällä "luokka/olio" esitystapaa (HL7-määrityksen tapauksessa useimmat luokat ovat "dataobjekteja", joihin ei olennaisesti kuulu luokan käyttäytymiseen liittyviä kohtia). Tässä vaiheessa HDF-menetelmä on kuitenkin selvästi vasta kehityksessä ja sisältää runsaasti epäselviä kohtia. Esimerkiksi otsikkotasolla puhutaan suhteista, triggereistä ja rajoitteista, mutta itse tekstissä ei.

HDF luvun 2 lopussa on seuraava kuvaus ja esimerkki toimintasäännöistä, jossa tosin ei kuvata suoraan onko kulloinkin kyseessä suhde, trigger tai rajoite:

Attached is an example of Business Rules, an output of the process step - Capture Business Rules: Relationships, Triggers, and Constraints.

Business Rules are declarative statements that enforce system behavior to comply with business policy and processes.

Business rules can be categorized depending on the expected target, information. Common types of rules include:

- **Information Integrity rules:** rules that state the conditions that an information element must meet to be considered valid for its purpose. An example could be as simple as “the date/time of discharge cannot be earlier than the date/time of admission”. Other information integrity rules may refer to the presence of related information or constrain the range of valid values from a vocabulary domain to meet a particular purpose. Information integrity rules help a system collect appropriate information to be used in later automated processes.
- **Information Accountability rules:** rules that state the circumstances that enable or constrain information collection, use and disclosure. “Who is allowed to see what information under what circumstances”. The rule may also specify that some accountability information must be recorded when information is added, changed or transferred and made available with the related information. Role Based Access Control is a formal way of expressing these rules, and is directly involved in determining security and audit requirements.
- **Business Process rules:** rules that identify the conditions that must be met before a specific task within a process can be started or ended. An example might be “a test result is not considered complete until the responsible professional has signed the result”.

The business rules that are captured during the Requirements Definition phase help to inform the specification modeling and specification documentation phases and are particularly useful for implementation mapping and testing to ensure systems integration is successful and meaning is preserved.

INPUTS REQUIRED:	<ul style="list-style-type: none"> • Documented Dynamic Behavior and Static Structure • Activity Diagram • Domain Analysis Model • Domain Glossary
PROCESS:	<ul style="list-style-type: none"> • Document Data Exchange Requirements using Data Objects in Activity Diagrams • Document the Dynamic Behavior/Interaction Requirements of an HL7 Specification - Conformance and Application Roles <ul style="list-style-type: none"> ○ Specifics of the interaction between the 'sending' and 'receiving' systems in the data/information-exchange-scenario-of-interest (i.e. the scenario that supports the originally identified business process.) ○ System-to-system interactions ○ Role Based Access Control
TOOLS:	<ul style="list-style-type: none"> • Activity Diagrams
RESPONSIBILITY:	<ul style="list-style-type: none"> • Domain Expert (A/R)
OUTPUTS:	<ul style="list-style-type: none"> • Documented Business Rules (See WP&D - 2.4.6)
NOTES:	<ul style="list-style-type: none"> • Although not discussed in this chapter, certain data/information structures may require additional analysis using UML state diagrams to clarify various stages of the class life cycle
WHEN:	<ul style="list-style-type: none"> • The generation of the artifacts mentioned above is not done sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done

5.2.5 Sovellusalueen analyysimallin harmonisointi (Harmonize the Domain Analysis Model with HL7 Reference Models)

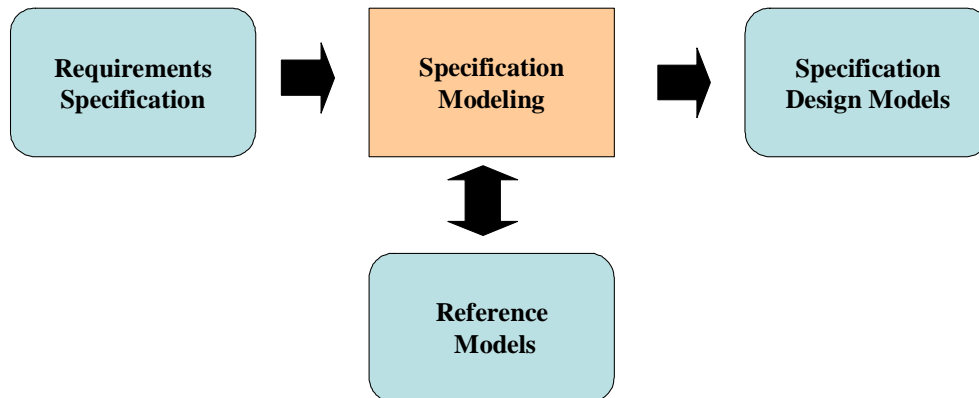
Vaativuotokset, jotka on saatu edellisissä vaiheissa, on harmonisoitava HL7-viitemallien kanssa. Tällä prosessilla varmistetaan, että useilta sovellusalueilta kerätyt ja sovellusalueen asiantuntijan ammattikielellä ilmaistut vaatimukset ilmaistaan yhdenmukaisella tavalla. Tammikuun 2006 jälkeen tulevissa ballot-paketeissa harmonisointia ei kuitenkaan tehdä vaatimusvaiheessa vaan vasta vaiheen 3 (Specification Modeling) jälkeen.

Harmonisointiprosessissa vaatimusmäärittelyn vaatimukset ilmaistaan käyttämällä HL7 viitemallin terminologiaa ja artefakteja. Eli jos vaatimusmäärittelyssä ei ole vielä käytetty HL7 viitemallia hyödyksi, on vaatimukset tässä kohdin muutettava HL7 viitemallin mukaisiksi. Kaikki harmonisointiprosessissa tunnistetut ristiriidat, päällekkäisyydet ja puutteet täytyy käsitellä. Tämän käsittelyn seurauksena voi olla vaatimusten päivittäminen HL7 viitemallin mukaiseksi. Tuloksena voi olla myös HL7 viitemallin laajentaminen tai tarkistaminen, jotta viitemalli mukautuisi niihin vaatimuksiin, jotka ovat jätetty pois tai jotka ovat puutteellisesti ilmaistu HL7 viitemallissa.

INPUTS REQUIRED:	<ul style="list-style-type: none"> • Documented Dynamic Behavior and Static Structure • Activity Diagram • Domain Analysis Model • Domain Glossary • HL7 Reference Models (See WP&D - 2.4.7) <ul style="list-style-type: none"> ○ HL7 Reference Information Model (RIM) ○ HL7 Data Type Specification (HDS) ○ HL7 Controlled Vocabulary Specification (CVS) ○ HL7 Domain Information Model (DIM)
PROCESS:	<ul style="list-style-type: none"> • The specifics of each data/information/interaction requirement must be ‘mapped’ to existing terminology and structure in HL7 Reference Models: • Domain Analysis Model & Domain Glossary: <ul style="list-style-type: none"> ○ associate domain-specific concepts with classes, attributes and/or vocabulary terms (i.e. look for equivalents/same definition) in the DIM; if an association does not exist, issue a change control, for a potential extension or enhancement to the DIM, into the modeling process ○ check the RIM; determine if a corresponding extension can be constructed; issue a change control, for a potential extension or enhancement to the RIM, into the modeling process • Storyboard and Activity Model: <ul style="list-style-type: none"> ○ Associate the set of interactions that the domain has created to published HL7 storyboards and corresponding interactions; ○ Is there potential to reuse (trigger events, message types, receiver responsibilities); create a list of interactions that can be reused; ○ net new information exchanges must be incorporated in the HL7 published storyboards; create interaction diagrams • Business Rules: <ul style="list-style-type: none"> ○ Examine the places where business rules (trigger events, class contributions, relationships) occur and determine if they already exist; if they do not exist, issue a change control for potential business rule change, into the modeling process • Update Requirement Specification Components as required: <ul style="list-style-type: none"> ○ Documented Dynamic Behavior and Static Structure ○ Activity Diagram ○ Domain Analysis Model ○ Domain Glossary • Apply accepted changes to HL7 Reference Models
TOOLS:	<ul style="list-style-type: none"> • Spreadsheet, database, or data mapping software • It is anticipated that much of this mapping will need to be supported with tools not yet defined (e.g. semantic network) or utilized
RESPONSIBILITY:	<ul style="list-style-type: none"> • HL7 Modeling Facilitator (A/R) • Domain Expert (R)
OUTPUTS:	<ul style="list-style-type: none"> • Harmonized Requirements Specification • A cross-reference between static requirements and HL7 reference models <ul style="list-style-type: none"> ○ An updated requirements specification document ○ A re-expression of the static requirements using structures from HL7 reference models ○ A collection of proposed enhancements to the HL7 reference models ○ Class Diagram • Interaction Diagrams • HL7 Reference Models Change Control Requests
NOTES:	<ul style="list-style-type: none"> • It is recommended that a harmonization meeting be convened to review and consider proposed changes to the HL7 Reference Models
WHEN:	<ul style="list-style-type: none"> • This process done sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done

5.3 Mallinnus (Specification Modeling)

Mallinnusvaiheessa vaatimusmäärittelyä käytetään viitemallin (reference model) muunnokseen suunnittelumalliksi. Vaiheen päätuotos on joukko määrittelyn tuloksena syntyneitä malleja (specification design models).



Mallinnukseen liittyy prosessi, jolla suunnitellaan HL7-määrittelyitä. Edellisen vaiheen vaatimusten harmonisoinnissa tuotetut vaatimusmäärittelyt ja harmonisoidut viitetietomallit ovat vaiheen 3 mallinnuksen ja suunnittelun syötteinä ennen siirtymistä paketoitiprosessiin ja hyödyntämiseen. Olemassa olevat määrittelyt aiemmista tai samanaikaisista suunnittelutoimista toimivat myös tämän prosessin syötteinä.

Mallinnuksen tuotoksia ovat

1. Staattiset tietomallit (DIM, tietotyyppien rajoitteet, sovellusalueen sanastojen (vocabulary) rajoitteet, CIM, serialized CIM),
2. Toiminnallisten näkymien mallit (sekvenssikaavio, yhteistyökaavio tilakaavio: tilat, tilasiirtymät ja näihin mahdollisesti liittyvät toiminnot),
3. Uudelleen käytettävät suunnittelumallien komponentit (DIM, muut uudelleen käytettävät suunnittelumallien elementit)
4. Yhteistyön ja interaktioiden suunnittelumallit

Osa suunnitteluprosessin tuotoksista on tarkoitettu äänestettäväksi itse standardiksi. Osa tuotoksista on tarkoitettu antamaan lisätietoa äänestettävänä olevaan standardiin ja eivät ole varsinaisen äänestuksen kohteena. Joitakin tuotetuista malleista on tarkoitus käyttää uudelleen myöhempien määrittelyjen osana. Kaikki määrittelyt on tuotettu vastaamaan vaatimusmäärittelyssä esitettyihin vaatimuksiin, ja ne käyttävät hyväkseen HL7-viitemalleja sekä aikaisemmin tehtyjä suunnittelutuotoksia.

Mallinnuksessa on seuraavat vaiheet:

1. Rakennetaan suunnittelumallit staattisiin tietojen näkymiin (DIM, RMIM, CIM, Serialized CIM)
2. Muodostetaan suunnittelumallit toiminnallisiin näkymiin (sekvenssikaaviot tietovirtojen (sanomien) havainnollistamiseen sovellusroolien välillä, yhteistyö- ja tilakaaviot suunnittelutyötä varten).
3. Määritellään uudelleenkäytettävät suunnittelumallien komponentit.
4. Muodostetaan yhteistyön ja interaktioiden suunnittelumallit (yhteistyökaaviot, sekvenssikaaviot)
5. Harmonisoidaan suunnittelumallit HL7:n viitemalleihin.

Alla kuvataan tarkemmin näitä vaiheita.

5.3.1 Staattisten tietonäkymien suunnittelu (Build Design Models of Static Information Views)

Sovellusalueen suunnittelumalli (DIM) peritään viitetietomallista (RIM). Se koostuu RIM-mallista kopioiduista luokista. DIM tarjoaa ratkaisun tietomalliksi tietyn sovellusalueen tiettyyn ongelma-alueeseen. Vertailemalla ja yhdistämällä sovellusalueen tietomallin vaatimukset RIM-malliin voidaan tunnistaa ne RIM-luokat, jotka pitää liittää DIM-malliin. Joissain tapauksissa voi olla tarve ottaa mukaan useita kopioita (klooneja) samasta RIM-luokasta. Jokaiselle RIM-mallista kopioidulle luokalle annetaan yksilöivä nimi, joka vastaa sen käyttöä toimintaprosessissa.

INPUTS REQUIRED:	<ul style="list-style-type: none"> • Requirements Specification (Domain Analysis Model) • HL7 Reference Models & Domain Information Models • Modeling Process Change Control
PROCESS:	<ul style="list-style-type: none"> • Create or Modify Static Model Design: <ul style="list-style-type: none"> ○ Create a Domain Information Model, if an appropriate domain information model does not exist - clone relevant classes from the reference information model and adjust the clone names, attributes, and relationships to reflect business rules and nomenclatures used in the requirements specification; multiple entry points ○ Modify a Domain Information Model, if an appropriate domain information model exists, modify / clone relevant classes from the HL7 Reference Information Model and adjust the clone names, attributes, and relationships to reflect business rules and nomenclatures used in the requirements specification; multiple entry points ○ Refine the Domain Information Model - incorporate restrictions on related data types and vocabulary domains from the Domain Analysis Model and incorporate references to design packages from earlier works; single entry point ○ Create Constrained Information Models; i.e. (a sub-set of the Domain Information Model) - package design information model clones in a logical fashion introducing new clones and additional refinements as needed ○ Create a Serialized Constrained Information Model, - apply annotations to the information model elements to enable the creation of a hierarchical data structure in those cases where serialization is an important aspect of the solution
TOOLS:	<ul style="list-style-type: none"> • V3 Static Model Design and Documentation Tools
RESPONSIBILITY:	<ul style="list-style-type: none"> • HL7 Facilitator (A/R) • HL7 Membership • Domain Expert (C)
OUTPUTS:	<ul style="list-style-type: none"> • Design Model of Static Information Views
NOTES:	<ul style="list-style-type: none"> • The development of the derived Static Information Models, is done through a process of restriction and constraint, not extension; this is the difference between HL7 software development methodology and UML software development methodology
WHEN:	<ul style="list-style-type: none"> • The generation of the artifacts mentioned above is not done sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done

5.3.2 Toiminnallisten näkymien suunnittelu (Construct Design Models of Behavioral Views)

Määrittelypaketin toiminnallisen osuuden suunnittelu toteutetaan jalostamalla vaatimusmäärittelyssä olevia aktiviteettikaavioita iteratiivisesti.

INPUTS REQUIRED:	<ul style="list-style-type: none"> • Requirements Specification (Activity Diagram & Business Rules) • Design Model of Static Information Views
PROCESS:	<ul style="list-style-type: none"> • Develop the Design Model of Behavioral Views: <ul style="list-style-type: none"> ○ Develop a collaboration model by refining the activity model; reflect the interaction of objects in the Activity Diagram; use a set of standard conventions regarding naming of objects and the depiction of inter-object linkages and associated message ○ Use the state transition diagramming to clarify the conditions that cause information to be exchanged and the expected information exchange behavior of a receiving system
TOOLS:	<ul style="list-style-type: none"> • UML Sequence Diagram (demonstrate message flow between application roles) • UML Collaboration Diagram (used for design work) • UML State Transition Diagram (used for design work)
RESPONSIBILITY:	<ul style="list-style-type: none"> • HL7 Facilitator (A/R) • Domain Expert (C)
OUTPUTS:	<ul style="list-style-type: none"> • Design Model of Behavioral Views (Sequence Diagram /Interaction Model)
NOTES:	<ul style="list-style-type: none"> • Business Rules are used to design the Behavioral Views
WHEN:	<ul style="list-style-type: none"> • The generation of the artifacts mentioned above is not done sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done

5.3.3 Uudelleenkäytettävien mallien määrittely (Define Reusable Design Model Components)

Sekä aiemmin muualla tehdyistä HL7-malleista että prosessissa luoduista malleista pyritään löytämään uudelleenkäytettäviä osia. Mallinnuksessa on oma vaihe tätä varten.

INPUTS REQUIRED:	<ul style="list-style-type: none"> • Requirements Specification • Design Model of Behavioral Views • Design Model of Static Information Views • HL7 Re-usable Design Models (Static Models) • HL7 Domain Information Models
PROCESS:	<ul style="list-style-type: none"> • Examine your Static Information Model, the existing HL7 reusable design models and the domain information models of related domains to see what design model components can be reused: <ul style="list-style-type: none"> ○ Find common modeling patterns ○ Consult with the stewards of the other domains; ○ Participate in a focus group to facilitate the alignment of the components of the models that can be reused ○ Joint design process because the results may require modification of multiple models
TOOLS:	<ul style="list-style-type: none"> • HL7 Static Model Designer
RESPONSIBILITY:	<ul style="list-style-type: none"> • HL7 Facilitator (A/R) • HL7 Membership (R) • Domain Expert (C)
OUTPUTS:	<ul style="list-style-type: none"> • Reusable Design Model Components
NOTES:	<ul style="list-style-type: none"> • This is a harmonization and negotiation process; recognition of a common problem with the intent to produce a common solution • The resulting reusable design pattern is the responsibility of one of the TSC and appears in their domain and is used by reference in the other TSC static models.
WHEN:	<ul style="list-style-type: none"> • Joint design is conducted through the following: <ul style="list-style-type: none"> ○ Working group meetings, held 3 times per year, scheduled teleconferences ○ Harmonization meeting, held 3 times per year between regular working group meetings ○ Special Purpose Project Meetings scheduled 30 days in advance and receiving Board Approval

5.3.4 Vuorovaikutuksen suunnittelu (Construct Design Models of Collaboration and Interaction)

Vuorovaikutuksen suunnittelussa pyritään kuvaamaan käyttäytyminen, joka vastaa storyboardin tavoitteita ja joka on yhdenmukainen määriteltyjen toiminnallisten näkymien ja tietosisältöjen kanssa.

INPUTS REQUIRED:	<ul style="list-style-type: none"> • Requirements Specification (Storyboard, Activity Diagram) • Design Model of Behavioral Views • Design Model of Static Information Views • Reusable Design Model Components • HL7 Reference Information Model (State Models)
PROCESS:	<ul style="list-style-type: none"> • Construct a collaboration diagram: <ul style="list-style-type: none"> ○ Determine the generalized system responsibilities for sending and receiving information (application roles) • Construct a sequence diagram: <ul style="list-style-type: none"> ○ Shows the set of interactions between the application roles in the sequence required to meet the objectives of the Storyboard ○ Examine the HL7 state models to ensure that you have appropriate information exchange at any critical state change of the focal class in your static model; refer to Activity Diagram & Design Model of Behavioral Rules to give you information on signals which would suggest the requirement for a trigger event ○ Define which application roles are responsible for question/answer type of interactions
TOOLS:	<ul style="list-style-type: none"> • UML Collaboration Diagram • UML Sequence Diagram
RESPONSIBILITY:	<ul style="list-style-type: none"> • HL7 Facilitator (A/R) • HL7 Membership (R) • Domain Expert (C)
OUTPUTS:	<ul style="list-style-type: none"> • Design Models of Collaboration (UML Sequence Diagram) • Definition of Trigger Events, Interactions and Application Roles
NOTES:	<ul style="list-style-type: none"> • The process is dependent on which type of interactions being designed • Question/answer type of interactions use the HL7 query mechanism; based on a static information model that may be designed for other operational purposes or it could be designed specifically for query results
WHEN:	<ul style="list-style-type: none"> • The generation of the artifacts mentioned above is not done sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done

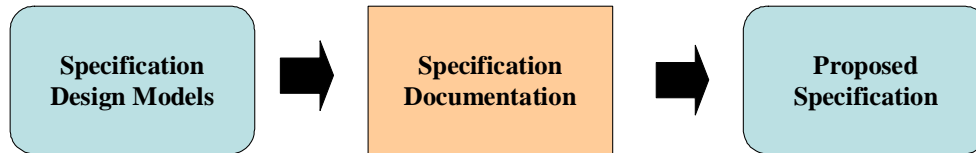
5.3.5 Suunnittelussa syntyneiden mallien harmonisointi (Harmonize Design Models with HL7 Reference Models)

Suunniteltujen mallien harmonisointi HL7:n viitemalleihin on sovellusalueesta vastuussa olevien teknisten komiteoiden yhteistyötä. Harmonisointi tapahtuu kahdella tasolla, sovellusalueen tasolla sekä laajennuksina HL7:n viitemalleihin.

INPUTS REQUIRED:	<ul style="list-style-type: none"> • Requirements Specification • Design Model of Behavioral Views • Design Model of Static Information Views • Reusable Design Model Components • Design Models of Collaboration and Interaction
PROCESS:	<p>Level 1 – Extensions of the Domain Information Model are required:</p> <ul style="list-style-type: none"> • Propose changes to the DIM, negotiate with the TC to have changes incorporated into the DIM <p>Level 2 – Extensions to the HL7 Reference Models are required:</p> <ul style="list-style-type: none"> • If a data type change is required, create change proposal, submit the proposal to the responsible TC committee; data types then go through the regular ballot process • If structural vocabulary or HL7 Reference Information Model changes are required, then the change proposals are made through the Harmonization Process, sponsored by TSC: <ul style="list-style-type: none"> ○ Meetings are held to obtain consensus to change the HL7 Reference Models ○ Facilitator prepares change proposal and then publishes it through the list servers in advance of the meetings ○ Steward Representative takes the vote; i.e. change accepted as proposed, change rejected; change accepted with agreed to modifications
TOOLS:	<ul style="list-style-type: none"> • Current and Proposed View of DIM or RIM & Rationale Cover Sheet • Vocabulary Content Spreadsheet & Rationale Cover Sheet
RESPONSIBILITY:	<ul style="list-style-type: none"> • HL7 Facilitator (A/R) • Domain Expert (C)
OUTPUTS:	<ul style="list-style-type: none"> • Reference Model Change Proposals • Design Information Models (Harmonized) • HL7 Reference Models (Harmonized)
NOTES:	<ul style="list-style-type: none"> • Tool set is currently being evaluated for changes subsequent to the tools being aligned with Model Interchange Format
WHEN:	<ul style="list-style-type: none"> • The generation of the artifacts mentioned above is not done sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done • Joint design is conducted through Working Group Meetings, held 3 times per year or Harmonization Meetings, held 3 times per year between regular working group meetings

5.4 Määrittelyn dokumentointi (Specification Documentation)

Määrittelyn dokumentoinnissa suunnittelumallit jaetaan loogisiksi paketeiksi, joihin on liitetty niitä selittävä teksti ja valmius hyväksyttäväksi. Tämän vaiheen päätuotos on hyväksyttäväksi ehdotettu määrittely.



1. Jaetaan suunnittelumallit loogisiksi paketeiksi.
2. Kirjoitetaan paketteihin selittävä teksti, esimerkit ja suunnittelun perustelut.
3. Päivitetään suunnittelumallit ja vaatimusmäärittelyt.
4. Kootaan ehdotettava määrittelypaketti.
5. Lähetetään määrittely hyväksyttäväksi.

5.5 Määrittelyn hyväksyminen (Specification Approval)

Määrittelyn hyväksymisessä ehdotettu määrittely joutuu käymään läpi hyväksyntävaiheita. Eri hyväksyntävaiheet voivat erota toisistaan riippuen hyväksyttävästä määrittelystä, hyväksymistasosta ja hyväksyttävän aihealueen kiinnostavuudesta. Tämän vaiheen päätuotos on hyväksytty määrittely.



1. Hyväksytetään määrittely äänestettäväksi teknisellä komitealla (TSC) ja hallituksella (board)
2. Muodostetaan äänestyspooli ja suoritetaan määrittelyn äänestys.
3. Arvioidaan vastustavat ja myönteiset kommentit.
4. Muunnetaan määrittelyä äänestyskommenttien perusteella.
5. Ratkaistaan vastustavien äänien kommentit ja jos tarpeen, toteutetaan uusintäänestys.

5.6 Määrittelyn julkistaminen (Specification Publication)

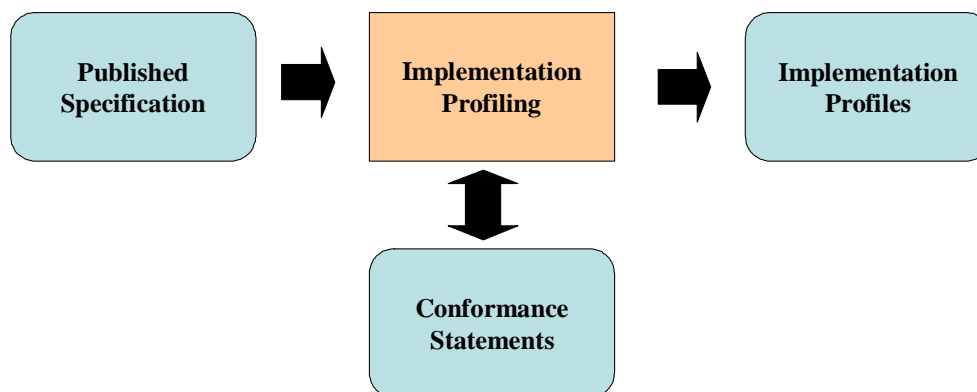
Määrittelyn julkistamisessa hyväksytty määrittely valmistellaan julkistettavaksi ja jaeltavaksi. Vaiheen päätuotos on julkistettu määrittely.



1. Pyydetään tekniseltä komitealta ja hallitukselta lupa määrittelyn julkistamiseen.
2. Valmistellaan määrittely julkistamista varten.
3. Lähetetään määrittelyn julkaisu standardointiviranomaiselle (ANSI/ISO).
4. Tehdään määrittelystä julkaisuja useille eri mediamuodoille.
5. Lähetetään ja jaetaan hyväksyttyä määrittelyä.

5.7 Toteutuksen ja käyttöönoton profilointi (Implementation Profiling)

Käyttöönoton profiloinnissa määrittelyn malleja jatkojalostetaan ja määrittelyä rajoitetaan tiettyjen suunnittelusääntöjen, käytäntöjen ja suositusten perusteella. Näin tuotetaan määrittelystä profiileja, joita käytetään tietyssä käyttäjien määrittelemässä ympäristössä. Käyttöönoton päätuotos on joukko määrittelystä tuotettuja profiileja ja vastaavuuslausumia (conformance statements).

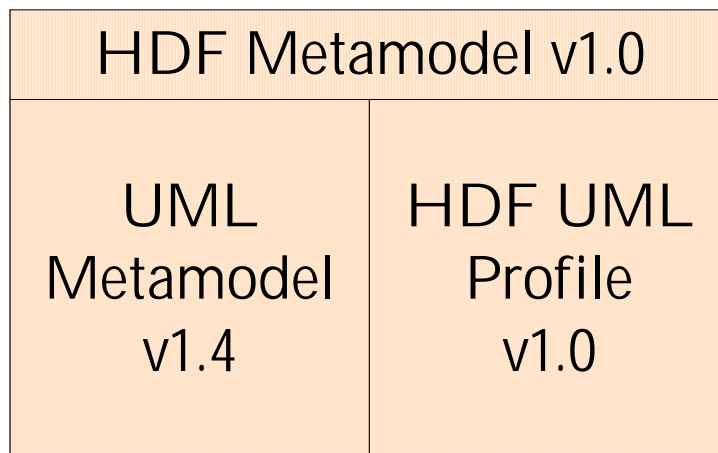


1. Tunnistetaan julkistetun määrittelyn käyttäjäkunta (community of uses)
2. Jatkojalostetaan ja rajoitetaan määrittelyn suunnittelumalleja.
3. Dokumentoidaan määrittelyyn kohdistuvat poikkeukset, laajennukset ja huomautukset.
4. Valmistellaan ja julkaistaan määrittelyn profiili.
5. Valmistellaan ja julkaistaan vastaavuuslausumat (conformance-määrittelyt).

6 HL7 UML Profile & Metamodel

HL7 V3 -kehitysmenetelmässä käytettävä mallinnussyntaksi perustuu UML:ään. UML:n semantiikka ja syntaksi kuvataan sen metamallissa. Siinä määritellään mallissa käytettävät elementit, niiden ominaisuudet ja suhteet sekä rajoitukset. HDF UML profiilissa (HDF UML Profile) määritellään HDF:n tarvitsemat laajennukset UML:n metamalliin. HL7:n laajennukset UML:ään on tehty käyttämällä UML:n laajennusmekanismeja: stereotyypit (stereotypes), tagien arvot (tag values) ja tagin arvon määrittelyt (tag value definitions). HDF metamalli muodostuu näin UML metamallista ja sitä laajentavasta HDF profiilista (kuva 7).

HDF Metamodel = UML Metamodel + HDF UML Profile



Kuva 7. HDF metamallin muodostuminen (Shakir 2002).

7 HDF Developers Guides

HDF Developers Guide -oppaat ovat HDF Methodology Specification -dokumentin tukidokumentteja. Ne on kohdistettu erityisesti standardien kehittäjille. Oppaat tarjoavat käytännön opastusta ja neuvoja HDF-kehitysmenetelmän käyttöön. Oppaiden on tarkoitus toimia HDF-prosessin olennaisimpien näkökulmien lähteenä ja opastaa standardien kehittäjiä läpi koko prosessin. Oppaita on tällä hetkellä kaksi:

- HDF Developer's Guide Vol 1 - Construction of a Design Level Static Model
- HDF Developer's Guide Vol 2 - Information Harmonization.

7.1 HDF Developer's Guide Vol 1 - Construction of a Design Level Static Model

Opas tarjoaa syvällisen kuvauksen prosessista, jossa rakennetaan suunnittelutason staattinen tietomalli. Suunnittelutason tietomallin rakentaminen tapahtuu HDF-menetelmän määrittelyn mallinnusvaiheessa (vaihe 3). Staattiset tietomallit sisältävät: sovellusalueen tietomallin (Domain Information Model, DIM), rajoitetun tietomallin (Constrained Information Model, CIM) ja lokalisoitun tietomallin (Localized Information Model, LIM). Staattisten tietomallien rakennusprosessi on pääpiirteissään samanlainen riippumatta tietomallista. Työkalut, joita käytetään staattisten tietomallien rakentamiseen, koostuvat joukosta makroja ja kaavioita. Makroja ja kaavioita käytetään laajentamaan Microsoft Vision sisäänrakennettuja ominaisuuksia.

Tässä oppaassa (Vol 1) kuvataan suunnittelutason staattisten tietomallien rakentamisprosessi; lähtien mallinnustyökalun lataamisesta HL7 web-sivulta ja päättyen staattisen mallin tallentamiseen suunnitteluvarastoon.

7.2 HDF Developer's Guide Vol 2 - Information Harmonization

Opas tarjoaa syvällisen kuvauksen prosessista, jossa harmonisoidaan tietomallit HL7:n viitetietomallin ja suunnittelutason tietomallien kanssa.

Tietomallien harmonisointia esintyy kolmessa HDF:n vaiheessa.

- Vaatimusten dokumentoinnissa (vaihe 2), jossa sovellusalueen analyysimalli harmonisoidaan HL7 RIM-mallin kanssa.
- Määrityksen mallinnuksessa (vaihe 3), jossa suunnittelutason tietomallit harmonisoidaan keskenään ja HL7 RIM-mallin kanssa.
- Käyttöönoton profiloinnissa (vaihe 7), jossa käyttöönoton tietomallit yhdistetään hyväksytyihin määrittelyihin ja profiileihin, tavoitteena on tunnistaa tietomallien poikkeukset ja laajennukset sekä huomautukset tietomalleihin.

Tässä oppaassa (Vol 2) kuvattu tietomallin harmonisointiprosessi on tarpeeksi yleinen, jotta sitä voidaan käyttää missä tahansa em. vaiheessa. Tietomallin harmonisoinnin tavoitteena on varmistaa,

että käyttäjän tietovaatimukset ja käyttöönotkokemukset heijastuvat HL7:n RIM-malliin ja suunnitelutason staattisiin tietomalleihin.

Lähteet

HDF Developer's Guide Vol 1 - Construction of a Design Level Static Model. 2004.

HDF Developer's Guide Vol 2 - Information Harmonization. 2004.

HL7 Development Framework. 2006.

Hinchley A. Understanding Version 3 - A primer on the HL7 Version 3 Communication Standard. Alexander Moench Publishing, Munich.

HL7 Training. 2002.

HL7 UML Profile. 2004.

HL7 Version 3 Ballot. 2004.

Shakir A-M. HL7 Development Framework Tutorial. 2002.

Shakir A-M. HL7 Development Framework Tutorial. 2003a.

Shakir A-M. HL7 Newsletter 04/2003. Health Level Seven, Inc. 2003b.