

HL7 Template Architecture -katsaus

Versio 1.0

	SerAPI-projekti	
	Yhteyshenkilö	Marko Suhonen (Marko.Suhonen@uku.fi)
	Dokumentin tila	Valmis
	Päiväys	4.7.2005

Sisällysluettelo

1	Johdanto.....	5
2	Template	6
2.1	Templaten käyttötarkoitukset.....	6
2.1.1	Archetype	6
3	Rajoitukset	7
3.1	Rajoitustavat.....	7
4	Templaten vaatimukset.....	8
4.1	Templaten esitys.....	8
4.2	Templaten tuottaminen	8
5	Templaten varastointi	10
5.1	Tietovarasto.....	10
5.2	Metatieto	10
6	Kielet	11
6.1	Object Constraint Language (OCL)	11
6.1.1	OCL-esimerkki	11
6.2	ADL (Archetype Definition Language).....	12
6.2.1	ADL-esimerkki.....	12
6.3	SR-XML: Templaten graafinen esitys.....	13
6.3.1	SR-XML -esimerkki	13
6.4	Web Ontology Language (OWL).....	14
6.4.1	OWL-esimerkki	14
7	Tulevaisuus	15
7.1	Template-case.....	15

Versiohistoria

Versio:	Pvm:	Laatijat:	Selitys:
Versio 1.0	4.7.2005	Marko Suhonen	Julkaisuversio

Esipuhe

Tämä työ liittyy SerAPI-hankkeeseen (Palveluarkkitehtuuri ja Web-sovelluspalvelut Terveydenhuollon 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 (päätos nro 40437/04) sekä joukko yrityksiä ja sairaanhoitopiirejä.

Dokumentissa kuvataan HL7 Template Architecturen keskeisiä piirteitä ja selvitetään mikä on template, sen käyttötarkoitukset ja ominaisuudet sekä keskeiset työkalut templatien toteuttamiseen.

Päälähteinä dokumentille ovat HL 7 V3 Standard Template Architecture -dokumentin versiot 3 ja 3.3.

1 Johdanto

HL7 Template Architecture (TA) on standardi, joka määrittelee syntaksin ja semantiikan kliinisten dokumenttien tai viestien komponenteille. Template on määritelty tieto-objekti, joka määrittelee rajoitteet R-MIM (Refined Message Information Model) -mallille tai sen osille (LIM). Template Architecturen päätarkoitus on määrittellä säännöt HL7 V3 Templaten tuottamisen, toteutuksen ja vertailun helpottamiseksi. Suomennoksena templatelle voidaan ajatella esimerkiksi rajoitusmallia, koska kyseessä on tiedon sääntöjen ja rajoitusten määrittelemine.

Dokumentin luvussa 2 kuvataan yleisesti, mikä on template ja sen käyttötavat sekä tarkoitukset. Luku 3 kuvaa Templaten asettamia rajoituksia ja luku 4 templaten ominaispiirteitä sekä keskeisiä näkökohtia templaten luomiseen. Luvussa 5 kerrotaan templateen liittyvistä tietovarastoista ja meta-tiedosta. Luku 6 kuvaa keskeiset työkalut templaten luomiseksi ja käyttämiseksi.

Huomioitavaa on, että Template Architecture on tämän dokumentin kirjoitusvaiheessa (kevät/kesä 2005) vasta kehitysasteella ja mitään tarkkoja määriytyksiä templaten rakentamisesta tai käyttökohteista ei tällä hetkellä ole tiedossa. Muutamia näkökohtia tulevaisuutta ajatellen on kuvattu luvussa 7.

2 Template

Yleisesti ottaen template on rakenteellinen kokoelma tietoa, jolla on merkitystä yhdelle tai useammalle terveydenhuollon osapuolelle. Templaten avulla HL7 mahdollistaa terveydenhuollon organisaatioille tiettyihin käyttötapauksiin tarkoitettujen template-joukkojen luomisen, ylläpidon ja levittämisen. Templaten avulla voidaan määritellä, mitä tietoa tarvitaan ja käytetään missäkin yhteydessä.

Template-joukkoja käytetään tiedon jakamiseen aliyhteisöissä organisaation tarkoituksen mukaisesti. Template-joukkojen käyttö viesteissä tai dokumenteissa edellyttää neuvottelua osapuolten kesken ja vaikuttaa siten sovelluskehitykseen. Templaten kehittäminen voi tapahtua esimerkiksi kahden yhteisiä intressejä omaavan organisaation välillä ja kehityksestä on tiedotettava HL7:lle. Tällaisille prosesseille tulee asettaa sääntöjä tulevaisuudessa.

2.1 Templaten käyttötarkoitukset

Templatella on useita käyttötarkoituksia. Templatea voidaan käyttää esimerkiksi ihmisten välisessä kommunikaatiossa ja rakenteellisen tiedon vaihdossa. Tietokoneiden välisten viestien validointi ja rajoitukset sisältävät viestin sisällön validoinnin templaten sääntöjen mukaan. Templatea voidaan käyttää tiedon syötössä, jolloin template toimii syöttöä ohjaavana välineenä määritellen esimerkiksi pakolliset ja vapaaehtoiset kentät sekä rajoitteet. Templatet ovat oliopohjaisia ja niitä voidaan määritellä rajoitusmallien perinnän avulla. Templatella voidaan määritellä ehtoja ja tätä kautta muun muassa päätöksentukea ja ohjelmallisia varoituksia tietyissä tilanteissa.

Templaten viestinkuljetus ja -käsittely mahdollistaa esimerkiksi tavan kuljettaa sopiva viestin osa käsiteltäväksi vastaanottavassa järjestelmässä. Templateja voidaan myös käyttää kuvaamaan tie-toelementtien välisiä suhteita ja niihin on mahdollista kohdistaa kyselyjä.

Perustavana kehyksenä templatelle toimii Reference Information model (RIM). RIM on tietomalli, joka kuvaa terveydenhuollon tietojärjestelmissä tarvittavat tiedot ja niiden väliset suhteet. Templatet ovat ryhmä hyvin määriteltyjä rajoitteita RIM-pohjaisille HL7 V3 -osille sisältäen Clinical Document Architecture (CDA). Rajoitteet ilmaistaan UML:n avulla ja rajoitetaan Object Constraint Language (OCL) -kielen avulla.

2.1.1 Archetype

Templatet voivat vaihdella paikallisista tietomalleista kokonaisuin kliinisiin dokumentteihin. Templatet edustavat syntaktisesti ja semanttisesti rakenteista tietoa, joka voi sisältää koodatun sanaston tai numeerista tietoa. Tietyt templaten alijoukot, jotka edustavat ydinkäsitteitä ja mahdollistavat Templaten rakentamisen, tunnetaan nimellä archetype ja ne ovat kliinisen informaation perusyksiköitä. Archetype-kirjasto ja sen edustamat määrittelyt sekä rajoitteet voidaan tuoda käyttöympäristöön. Voidaan sanoa, että template koostuu archetypeistä ja archetype on template.

3 Rajoitukset

Template on rajoite normaaliin HL7 V3 -määritykseen nähden. HL7 V3 -ilmentymien tulee vastata HL7 V3 -määrityksiin ja lisärajoituksia voidaan asettaa HL7 V3 Templaten avulla. Yksinkertaisesti tällä voidaan tarkoittaa esimerkiksi haluttujen käsitteiden poimimista viestistä sekä määrittellä samalla, poimitaanko tieto tekstinä, numeerisena tietona vai koodattuna.

3.1 Rajoitustavat

Templatella voidaan määrittää rajoituksia, jotka voidaan jaotella staattisiin, yhteisesiintymärajoituksiin, periytymisen rajoituksiin ja loogisiin rajoituksiin.

Staattisia rajoituksia ovat esimerkiksi kloonin ensisijaisten yhteyksien ja attribuuttien rajoittaminen, sallittujen aika- tai päivämäärämuuttujien syöttäminen, numeeristen ja merkkijonomuuttujien rajoittaminen sekä terminologian sallittujen koodien rajoittaminen.

Yhteisesiintymärajoitukset tarkoittavat käytännössä jonkin arvon rajoittamista toisen arvon perusteella, esimerkiksi periaatteella "jos arvo1 on X, on arvo2:n oltava Y". Rajoittaminen pätee muun muassa aika-arvoihin, numeeriseen vertailuun ja merkkijonomuuttujiin.

Templaten osat voivat olla järjestettyjä tai järjestämättömiä. Järjestetyssä templatessa rajoitusten asettamisjärjestyksellä on merkitystä, kun taas järjestämättömässä ei.

Yhdenmukaisuusprofiilin (conformance profile) ja templatien välinen ero on, että yhdenmukaisuusprofiili määritellään jo suunnitteluaikana, koko määrittämisestä koskien, ja se perustuu laukaisevan tapahtuman (trigger) synnyttämään viestiin, kun taas templatien käyttö tapahtuu triggereistä välittämättä.

4 Templaten vaatimukset

Templatella on seuraavat piirteet:

- Pysyvyys: Template on olemassa muuttumattomassa tilassa ajanjakson, joka määräytyy paikallisten sääntöjen määräämien vaatimusten mukaan.
- Templatea ylläpitää organisaatio, jonka hallintaan kyseinen template on annettu.
- Templaten ilmentymä on kokoelma informaatiota, joka on tarkoitettu tunnistettavaksi.
- Template määrittelee oletusarvoisen kontekstin sisällölleen.
- Kokonaisuus: Templaten ilmentymän tunnistaminen pätee koko dokumenttiin.
- Templaten ilmentymä on ihmisten luettavissa ja ymmärrettävissä.
- TA:t voidaan kuvata XML-muodossa.
- TA saa merkityksensä HL7 Reference Information Model:sta (RIM), DIM:sta, R-MIM:sta ja R-MIM -osista sekä käyttää HL7 V3 tietotyyppejä.
- TA-määrittely on joustava.

4.1 Templaten esitys

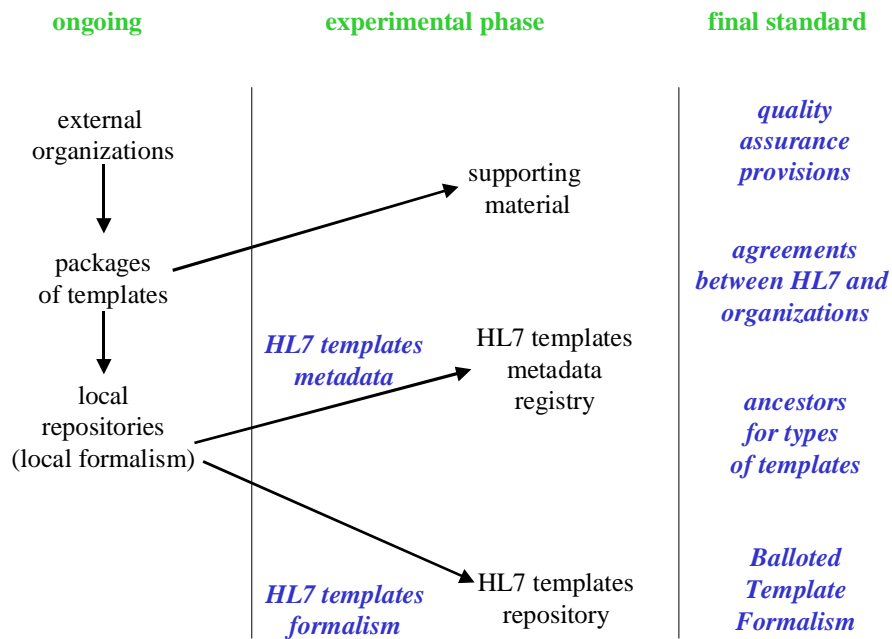
Templaten esitys on määriteltävä tavalla, joka sallii:

- Asiantuntijoiden määrittelemän kliinisen informaation tarkan syntaktisen ja semanttisen esityksen.
- Ilmaisun metakielellä, joka on alustariippumaton, mutta muunnettavissa jollekin tietylle tekniselle kielelle tai ohjelmistoalustalle.
- Esityksen kuvaamisen HL7 RIM -mukaisiksi osiksi ja käytännöiksi.
- Kuvaamisen UML-standardin mukaisena esityksenä sallituilla OCL-rajoitteilla.

Templaten tunnistamismekanismin tulee olla ITS (Implementable Technology Specification) -riippuvainen. On odotettavissa, että tieto saattaa kulkea usean ITS:n kautta ennen saapumistaan määränpäähän. Mekanismin tulee tukea templateja, joille voidaan määritellä toiminto, entiteetti, rooli tai osallistuminen. Tapa, jolla HL7 tunnistaa mikä viesti lähetetään, tulisi olla tiedossa kommunikaatiokääreeseen annetulla attribuutilla. Tällöin rakenteiset luokat, joiden on tarkoitus kuljettaa kliinistä tietoa, kuljettaisivat myös vaaditun templaten metatiedon.

4.2 Templaten tuottaminen

Asteittainen prosessi templaten tuottamiseen, ylläpitoon ja levitykseen esitetään kuvassa 1.



Kuva 1: Templaten luominen

HL7 SIG:iin (Special Interest Group) tai TC:hen (Technical Committee) voitaisiin sisällyttää keskeisiä templateja. Mikäli tarpeellista, tiettyjä templateja voidaan sisällyttää myös osaksi HL7-standardia tai mainita niistä. HL7 TC voisi päättää kehittää yhden tai useampia yleisiä templateja, joita voitaisiin käyttää "esi-isinä" muille templateille ja hyödyntää uusien templatien luomisessa.

5 Templaten varastointi

Templaten säilytystä varten on rakennettava tietovarasto ja/tai rekisteri, jossa templateja säilytetään. Tietovarasto voi olla esimerkiksi HL7:n palvelimelle sijoitettu.

5.1 Tietovarasto

HL7 tulee luomaan yleisen joukon metatietoa, jota käytetään luetteloimaan sekä templateja että yhdenmukaisuusprofiileja, ja joita säilytetään tietovarastossa (repository). Käyttäjä voi hakea templatea metatiedon kenttien avulla ja kahden templatien hierarkkinen suhde on ilmaistavissa, samoin templatien vertailu on mahdollista. Jokaisella templatella on metatiedolla toteutettu globaali ja yksilöllinen tunnistus, jolloin muuta tietoa ei tarvita templateen viittaamiseksi.

Template architecture voi käsittää tuhansia templateja, joiden keskinäiset suhteet on pidettävä selvinä, jotta käyttäjä voi helposti löytää tarvitsemansa templatien.

Standardin tulevat osat tulevat määrittelemään tapoja paikallisten tietovarastojen hallintaan ja selvittämään tarkemmin tietovarastojen yhteistoimintaa sekä lopulta tuottamaan mahdollisesti keskitetyn tietovaraston. Varastossa olevien templatien on noudatettava laadunvarmennusprosessia. Templaten toimittaja on tunnistettava ja templatien tuottamiseen on oltava sääntöjä koskien mm. mitkä templatet ovat muokattavissa jne.

HL7:n Template-kirjastoon kuuluu yhteystaulu, jossa osoitetaan mitkä HL7-määritykset sisältävät tarkoitetut komponentit. Käyttäjän luodessa template saadaan siis selville mihin määrityksiin se on sovellettavissa. Kirjastoon voidaan tehdä kyselyjä, joilla selvitetään mille HL7:n määrittelyjen komponenteille on mahdollista luoda template. Templaten luominen kerää metatietoa templatesta ja osoittaa myös tiedon rajoitetuista komponenteista.

5.2 Metatieto

Templaten metatiedon on sisällettävä muun muassa ID-tunnus, nimi, templatien kuvaus ja laajuus (pakollinen), rekisteröivä taho, esim. organisaatio, yhteisö tai henkilö (pakollinen); versionumero, luontipäivä ynnä muuta. Näitä tietoja voidaan hyödyntää muun muassa templatien hakemisessa, tekijän ja vastuullisen tahon hakemisessa, versioinnissa, ajankohtaisuudessa ja käyttötarkoituksissa.

Esimerkkinä tiedon varastoinnista mainittakoon ebXML, joka on kasvava kansainvälinen standardi liiketoimintakommunikaatioon Internetissä. EbXML-standardiin kuuluu metatietorekisteri, joka pohjautuu ISO 11179 metatietorekisteri-standardiin. EbXML-pohjaisia artefakteja on toteutettu OWL-kielen avulla. (Tarhonen)

6 Kielet

Tekniset spesifikaatiot (UML, XML ja SR-XML, OWL, ADL) sisältävät jaetun esityksen XML-yhteensopivilta osiltaan. Kieliä on käytetty eri tavoin (vertaa luku 5.2), mutta virallista tapaa ei ole tämän dokumentin kirjoitusvaiheessa (kevät 2005) varsinaisesti määritelty. Tietyvästi ADL-kieltä on sovellettu CDA:han ja viralliseksi template-kieleksi on kaavailtu OCL-kieltä, jota ei ilmeisesti aiota kuitenkaan soveltaa käytännössä. (Tarhonen)

- ADL tarjoaa kompaktin syntaktisen ja semanttisen esityksen kliinisestä informaatiosta (joka mappautuu UML:ksi OCL-rajoituksilla)
- OCL (GELLO-lisäyksin) tarjoaa standardin rajoituskielen UML:llä ilmaistuille HL7 RIM -elementeille Web-yhdenmukaisilla työvälineillä toteutettavaa esitystä varten (W3C standardit)
- SR-XML tarjoaa jaetun graafisen esityksen kliinisestä informaatiosta kliinisen alan asiantuntijoille, informaatioille ja ohjelmoijille (käännös XML, XSD, XSLT, OWL jne.).
- OWL, standardi verkkopohjaiseen ontologiseen esitykseen, tarjoaa semanttisesti monipuolisen esityksen kliinisestä informaatiosta.

6.1 Object Constraint Language (OCL)

OCL määrittelee tyypillisesti invariantteja ehtoja, joiden on pädeävä mallinnettavaan järjestelmään. Huomioitavaa on, että OCL-ilmausten arviointi ei aiheuta sivuvaikutuksia, jolloin niiden arviointi ei voi muuttaa vastaavan ajavan järjestelmän tilaa. Lisäksi invarianttien määrittelyssä UML-metamalliin voidaan käyttää OCL:ää sovelluskohtaisten rajoitusten ilmaisemiseen sekä hyvinmuodostuneisuuden sääntöjen määrittelyyn semantiikassa.

UML-kaaviolla ei yleensä voida määrittellä kaikkia olennaisia määrittelyn osia, jolloin OCL:ää käytetään lisärajoitusten määrittelemiseen tilanteissa, joihin luonnolliset kielet ovat sopimattomia. OCL on muodollinen kieli, jota on helppo lukea ja kirjoittaa. OCL ei ole ohjelmointikieli, joten sillä ei voida toteuttaa esimerkiksi ohjelmalogiikkaa.

OCL:ää voidaan käyttää esimerkiksi luokkien, tyyppien ja stereotyyppien invarianttien määrittelyyn, metodien tai operaatioiden esi- ja jälkiehtojen määrittelyyn, navigaatiokielenä ja operaatioiden rajoitusten määrittelyssä sekä usein myös semantiikan säännöissä.

OCL on tyyppitetty kieli, joten jokaisella OCL-lauseella on tyyppi. OCL-lauseen on oltava sääntöjen mukainen, eli esimerkiksi lukuarvojen ja merkkijonojen vertailu on mahdotonta.

6.1.1 OCL-esimerkki

Sana *context* tarkoittaa lauseen kontekstia. Sanat *inv*, *pre* ja *post* ilmaisevat stereotyypit, vastaavasti rajoituksessa «invariant», «precondition», ja «postcondition». Itse OCL-lause tulee kaksoispisteen jälkeen.

```
context TypeName inv :  
'this is an OCL expression with stereotype <<invariant>> in the
```

context of TypeName' = 'another string'

6.2 ADL (Archetype Definition Language)

ADL on templatien ja archetypejen mallintamiseen tarkoitettu kieli. ADL:n syntaksi on rakenteinen, hieman samaan tapaan kuin lohkorakenteisissa ohjelmointikielissä. ADL ottaa huomioon archety-
pen elinkaaren näkökulmat.

ADL:n syntaksi käsittää keskeisimpinä:

- Archetyphen tunnistaminen
- Käännös luonnollisille kielille
- Suhteet terminologioihin
- Archetyphen ja templatien suhde

On vielä päättämättä, tuleeko archetyphen tunniste sisältämään tiedon elinkaaren tilasta ja miten se esitetään.

6.2.1 ADL-esimerkki

Lyhyt esimerkki ADL:n syntaksista,

```
header_part
    archetype identification, meta-data
body_part
    archetype                                     definition
    terminology_part
    terminology definitions and binding
```

jossa lihavoidut sanat ovat avainsanoja (keyword).

Header-osan syntaksi noudattaa seuraavaa muotoa:

```
archetype                                     some.archetype.id
concept                                       some_term
[ specialise some.other.archetype.id ]
description
purpose: "xxxxx"
use: "xxxxx"
submitter: "xxxxx"
author: "xxxxx"
status: "xxxxx"
other_meta_data_item: "xxxxx"
```

6.3 SR-XML: Templaten graafinen esitys

SR-XML:n (Structured Reporting-Extended Markup Language) suoraviivainen graafinen esitys mahdollistaa tiedon esittämisen kliinisten alueiden asiantuntijoiden erikoisosaamisen mukaan.

Kliininen informaatio on toinen kahdesta perustavasta templatesta, jotka sisältyvät yleisiin kliinisiin dokumentteihin. Tämä voidaan esittää graafisesti SGML:n Elm-puun avulla, ja mallintaminen DTD:lle tai XML schemalle on mahdollista. Samoin muunnos ADL:lle (luku 6.2) ja OWL:lle (luku 6.3) OCL:n rajoittein on toteutettavissa. Tämä sallii muodollisen kliinisen syntaksin ja semantiikan sekä liittyvien HL7-osien rajoituksen esittämisen.

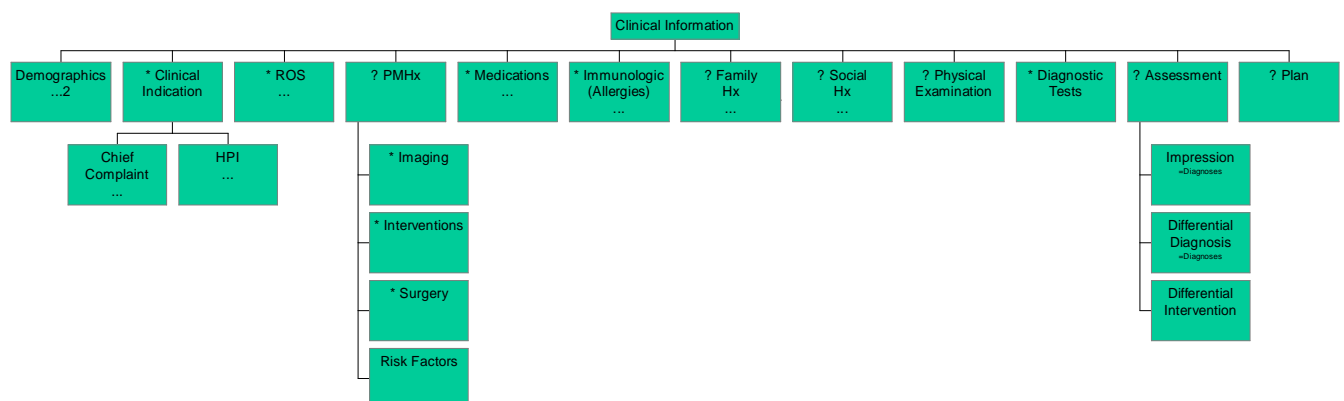
6.3.1 SR-XML -esimerkki

Esimerkissä kuvataan SR-XML -notaation mukainen Elm-puu, joka esittää kliinistä informaatiota.

Taulukossa kuvataan kaavion sisältämien symbolien merkitys.

Taulukko 1: SR-XML -symbolit

Symboli (XML)	Tarkoitus
	Elementtien järjestys
*	0 tai enemmän
+	1 tai enemmän
?	0 tai 1 (vaihtoehtoinen)
#PCDATA	merkkitieto
=	yhtä kuin
...	jatkuu muissa templateissa



Kuva 2: SR-XML, kliininen informaatio

6.4 Web Ontology Language (OWL)

Tiedon esityksen entiteetissä (Knowledge Representation) luodaan ensimmäisenä OWL-ilmentymä CDA R2 -mallista.

OWL on ontologiakieli, jota voidaan käyttää verkkodokumenttien terminologian tarkoituksen kuvaamiseen. OWL tukee luokkamäärittäjiä ja perintää sekä määrittäjiä hierarkkisuuksiin, arvojen rajoituksiin, totuusarvoihin ja useisiin muihin käsitteisiin.

6.4.1 OWL-esimerkki

Lyhyt esimerkki OWL-kielestä. Esimerkki kuvaa henkilön syntymäpaikkaa.

...

```
<owl:Class rdf:ID="Birthplace">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#has_birthplace"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:about="#Place"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#CDA_level2"/>
</owl:Class>
```

...

7 Tulevaisuus

Tulevaisuus osoittaa templatien hyödyllisyyden ja tarkoituksenmukaisuuden. Tällä hetkellä olemassa on HL7 Templates Committee (TEMPLATES) -järjestön päätöksentekokäytäntöihin liittyvä dokumentti sekä ajatuksia templatien ja archetypejen yhteentoimivuuden toteuttamiseksi openEHR:n, CEN:in ja HL7:n välillä. Aikataulu projektin toteuttamiseksi on luettavissa osoitteessa <http://www.hl7.org/Library/Committees/template/Template%20Archetype%20Alliance%20Project50505timeline%2Edoc> ja tämän dokumentin mahdolliset tulevat versiot tulevat viittaamaan projektin tuotoksiin.

Lähinnä toteutusta oleva taso on CEN:n määrittelemä (tämän dokumentin kirjoitushetkellä kesken-eräinen) "**Error! Reference source not found.: Error! Reference source not found.**" - dokumentti, jossa kuvataan erinäisiä archetypejen määrittelytapoja ja perusteita archetypejen vaihtoon rekisterien välillä. Edellä mainitun dokumentin seuraava versio tulee kuvaamaan menettelyjä tarkemmin. Luvussa 7.1 kuvattu esimerkki kertoo HealthVetin näkemyksen templatien käyttämisestä.

7.1 Template-case

HealthVet ja siihen liittyvä Veterans Health Administration (VHA) määrittelevät tietomallin VHA Health Information Model (VHIM), joka edustaa kaikkea jaettavissa olevaa tietoa VHA:n sisällä.

VHIM on suuri integroitu malli ja sisältää paljon tietoa, jota ei tarvita kaikissa projekteissa. Odotettavissa on, että projektit käyttävät vain sitä osaa tiedosta, joka on olennaista tiettyä käyttötapausta varten. Tällöin tarvitaan oleellisen tiedon alijoukkojen tunnistamista ja käyttöä templatien avulla.

VHA Template Registry tulee olemaan näiden alijoukkojen tietovarasto. Jokainen template tukee yhtä tai useampaa käyttötapausta. Templatet ovat uudelleenkäytettäviä rekisteröinnin ja versioinnin kautta, jolloin esimerkiksi muut projektit voivat käyttää templatea.

Templatet ovat tuotteita, joita voidaan käyttää ja ajaa suoraan ohjelmistoissa. Tämä on myös vaatimus, koska templatien on oltava valmis käytettäväksi. Templatet kehitetään edelleen Java-luokiksi ja XML-skeemaksi, jolloin mallista periytyvä koodi saadaan suoraan käytettävään muotoon. Etuna tässä on useiden tuotteiden tuki, jolloin VHA-templatet ovat tekniikkariippumattomia. Sovelluksissa nämä tuotteet voivat toimia esimerkiksi sisäisinä tietorakenteina tai käyttöliittymäkomponentteina. Tieto voidaan tällöin esittää templatien ilmentymän avulla suoraan käyttäjälle tai sitä voidaan käyttää sovelluksessa tiedon vastaanotossa. Muita käyttökohteita ovat esimerkiksi palvelurajapinnat (mm. kyselyiden parametreina), kyselyt ja valinnat.

HealthVet:n palvelurajapinnat paljastetaan Cross-Application Integration Protocol (CAIP) -protokollan avulla ja tämä mahdollistaa useiden väliohjelmistojen käytön. Rajapintaa käytettäessä yksi parametreista on template. Mallissa ja infrastruktuurissa ei (välttämättä) käytetä HL7-viestejä - varsinkaan sisäisessä toteutuksessa - mutta järjestelmien välinen viestintä tapahtuu HL7-viestien mukaisesti. HealthVet näkee tämän etuna, koska sisäisiä viestejä voidaan käyttää omassa muodossaan ja järjestelmästä ulos kulkevat viestit muunnetaan HL7-muotoon väliohjelmiston tai gatewayn avulla. (Rubin)

Lähteet

Elkin, Peter L.; Kernberg, Martin. HL 7 V3 Standard Template Architecture 3.3.

<http://www.hl7.org>, 27.2.2005. Viitattu 22.4.2005.

<http://www.hl7.org/Library/Committees/template/HL7v3templates33feb282005M1%2Ezip>

Elkin, Peter L.; Kernberg, Martin. HL 7 V3 Standard Template Architecture 3.0, 8.12.2004. Viitattu 18.4.2005.

Tarhonen, Timo. Sähköposti. 11.5.2005. Viitattu 12.5.2005.

Rubin, Ken. A HealthVet Software Architectural Walkthrough. 27.4.2005. Viitattu 17.5.2005.