
Terveystenhoidon avoimet sovellusrajapinnat - koodistörajapinnat

 HL7 Finland ry	HL7 Finland ry, SerAPI-projekti, PlugIT-projekti OID: 1.2.246.777.11.2005.12 Ydinpalvelusrajapinnat Yhteyshenkilö: Juha.Mykkanen@uku.fi
	Versio 2.0 - Hyväksytty HL7 Finland hallituksessa 10.5.2005 kommentointikierroksen 3.12.2004 - 7.1.2005 sekä lausuntokierroksen 4. - 25.2.2005 jälkeen
	viimeksi päivitetty: 18.5.2005

Tiivistelmä

Terveysthuollon ohjelmistojen sisältämien tietojen luotettava siirto sovellusten välillä tai yhteismitallisen tiedon tuottaminen eri järjestelmistä vaatii sitä, että sovellusohjelmistot voivat jakaa ja käyttää samoja koodistoja ja terminologioita erityisesti rakenteisen tiedon yhteydessä. Koodistojen epäyhteensopivuus, paikalliset ja sovelluskohtaiset koodistot sekä samojen koodistojen epäyhtenäiset versiot aiheuttavat runsaasti yhteensopivuusongelmia sovellusohjelmistojen välillä terveydenhuollossa.

Tässä raportissa esitetään taustatiedot, vaatimukset, tekniikkariippumattomat ratkaisut ja tekniset tarkennukset avoimille ohjelmistorajapinnoille keskitettyjen koodistopalveluiden käyttämiseksi. Rajapintoja toteuttamalla voidaan mm. vähentää päällekkäisen tiedon syöttöä eri järjestelmiin, päällekkäistä koodistojen ylläpitoa eri sovelluksissa, saada käyttöön samoja koodistoja ja yhtenäisiä koodistojen versioita eri ohjelmistoissa ja tuottaa uudelleenkäytettäviä ohjelmistopalveluita.

Raportin ratkaisulla voidaan tukea ja täydentää Stakesin Kansallisessa koodistopalvelussa ylläpidettävien koodistojen käyttöönottoa sovellusohjelmistoissa. Määritellyt rajapinnat perustuvat kansainvälisten rajapintastandardien ja Kansallisen koodistopalvelun tietosisällön hyödyntämiseen. Rajapinnat ovat riippumattomia käytetyistä koodistoista, ja myös terveydenhuollon ulkopuolisia koodistoja voidaan hyödyntää rajapintojen avulla.

Raportti on tuotettu PlugIT-hankkeessa tutkijoiden, terveydenhuollon organisaatioiden ja ohjelmistoyritysten yhteistyössä vuosina 2003-2004. Sen tuottamisessa on hyödynnetty projektin integroitiratkaisujen arviointi- ja määrittelymenetelmiä.

Raportti julkaistiin alun perin julkaisusarjassa ”PlugIT-hankkeen selvityksiä ja raportteja”:
(<http://www.plugin.fi/julkaisut/tilauslomake.html>):

Juha Mykkänen, Saara Savolainen, Hannu Virkanen:

Terveysthuollon avoimet sovellusrajapinnat: koodistorajapinnat.

PlugIT-hankkeen päättymisen jälkeen dokumentti läpäisi HL7 Finland:in kommentointi- ja lausuntokierrokset, ja HL7 hallitus hyväksyi sen 10.5.2005. Verrattuna PlugIT-julkaisuun, tähän dokumenttiin on lisätty kommentointi- ja lausuntokierrosten kommentit (liite 5), eri osien normatiivisuutta havainnollistava kappale lukuun 1.4 ja HL7:n OID-dokumenttitunnus. Dokumentin muut osat ovat samoja kuin aiemmin julkaistussa raportissa.

Esipuhe

Tämä selvitys on tehty PlugIT-hankkeessa vuosina 2001-2004. Selvitys dokumentoi PlugIT-hankkeessa tuotetut koodistojen avoimet sovellusrajapinnat taustatietoineen ja vaatimuksineen. Selvitys on suunnattu koodistojen hyödyntämisen kehittämiseen osallistuville terveydenhuollon ja tietohallinnon ammattilaisille, ja ohjelmistojen tuottajille. Selvityksen alussa esitetyt tarpeet, vaatimukset ja perusratkaisut ovat hyödyllisiä esimerkiksi kansallisesti ylläpidettävien koodistojen hyödyntämistä suunniteltaessa. Selvitys sisältää myös tarkat rajapintamäärittelyt, joiden avulla terveydenhuollon sovellusten toteuttajat voivat tuottaa tai hyödyntää avoimia koodistorajapintoja.

PlugIT-hanketta ovat rahoittaneet ja siihen ovat osallistuneet TEKES, Mawell konserni, Medimaker Oy Ltd, Medici Data Oy, Mediweb Oy, Mylab Oy, Tietoenator Oyj, WM-data Novo Oyj, Atkos Oy, BEA Systems Oy, Commit Oy, Enfo Oy, Fujitsu Services Oy, General Electric Healthcare CIS EMEA, Mediconsult Oy, Microsoft Oy, Oracle Finland Oy, Helsingin ja Uudenmaan sairaanhoitopiirin kuntayhtymä, Pirkanmaan sairaanhoitopiirin kuntayhtymä, Pohjois-Savon sairaanhoitopiirin kuntayhtymä, Pohjois-Pohjanmaan sairaanhoitopiirin kuntayhtymä, Satakunnan sairaanhoitopiirin kuntayhtymä, Varsinais-Suomen sairaanhoitopiirin kuntayhtymä, Kuopion kaupungin sosiaali- ja terveystakeskus sekä Siilinjärven ja Maaningan terveydenhuollon kuntayhtymä.

Tekijät kiittävät projektiin osallistuvien yritysten ja sairaaloiden edustajia ja kaikkia koodisto- ja ydinrajapintatyöhön osallistuneita projektin työntekijöitä runsaista ja hyvistä ideoista ja kommentteista.

Kuopiossa 30. elokuuta 2004

Tekijät

Sisällys

1 JOHDANTO	6
1.1 PlugIT-projekti	6
1.2 Sovellusten koodistorajapintojen määrittelyt	6
1.3 Dokumentissa käytettyjä käsitteitä	8
1.4 Raportin rakenne	9
2 TAUSTASELVITYKSET	10
2.1 Nykytila ja kehittämiskohteet	10
2.2 OMG Terminology Query Services (TQS)	11
2.3 HL7 version 3 koodistojen tietotyypit	20
2.4 HL7 Common Terminology Services (CTS)	20
2.5 Stakesin koodistopalvelin	23
2.6 OID-yksilöinti	24
3 INTEGROINTIVAATIMUKSET	27
3.1 Tavoitteet ja päärajaukset	27
3.2 Osallistuvat ohjelmistot ja sovellusaluevaatimukset	28
3.3 Koodistopalvelut	29
3.4 Tietosisältö	30
3.4.1 Minimitiedot	30
3.4.2 Monikieliset koodistot, hierarkian käsittely ja edistyneet haut	30
3.4.3 Vapaa tietosisältö	31
3.5 Vaatimusten tarkempi kuvaus	31
4 TEKNIKKARIIPPUMATTOMAT LIITTYMÄMÄÄRITTELYT	38
4.1 Integroitiperiaatteet ja sovellusten vuorovaikutus	38
4.2 Tekniikkariippumattomat operaatiomäärittelyt	40
4.2.1 Minimitaso	41
4.2.2 Perustaso (base)	43
4.2.3 Monikieliset koodistot (multilingual)	46
4.2.4 Vapaa tietosisältö (freeElements)	47
4.2.5 Edistyneet haut (advSearch)	49
4.2.6 Hierarkkisten koodistojen käsittely (hierarchical)	50
4.2.7 Paikalliset koodit ja koodien tila (status)	52
4.3 Muut tekniikkariippumattomat määrittelyt	53
4.3.1 Yleiset virhetilanteet	53
4.3.2 Koodistohyödyntäjä-esimerkkejä	53
4.4 Integroititekniikoiden vaatimukset	54
4.5 Toteutuksiin kohdistuvat vaatimukset	55
4.5.1 Toteutuskohtaisiksi jätettävät seikat	55
4.5.2 Määrittelyn mukaisuus	56
4.6 Määrittelyn käyttö ja jatkokehitys	57
4.7 Tekniikkariippumattomiin määrittelyihin tehdyt muutokset	58
5 TEKNISET LIITTYMÄMÄÄRITTELYT XML- JA HTTP-RAJAPINNOILLE	59
5.1 Tavoitteet ja päärajaukset	60

5.2	Aikaisempien määritysten tarkentaminen	61
5.2.1	Tekniikoihin kohdistuvat vaatimukset	61
5.2.2	Rajapintojen ja tasojen yhdistäminen	62
5.3	Valitut tekniikat ja niiden käyttötavat	62
5.4	Tekniset sovellusosat, niiden vastuut ja tiedonvälitys	62
5.5	Operaatio- ja parametrimääritykset	63
5.5.1	Määrittelyn palvelutasot ja niillä vaaditut piirteet	64
5.6	Codeservice-rajapinta	65
5.6.1	GetSupportedCodeSystems - koodistotarjoajan tukemat koodistot	65
5.6.2	GetSupportedServices - koodistotarjoajan palvelut	66
5.6.3	GetInfo - koodistotarjoajan hallinnolliset ja yhteystiedot	67
5.7	Codeset-rajapinta	68
5.7.1	LookupCodesByDesignation - koodien haut selitteillä	68
5.7.2	ListCodes - koodien listaus	71
5.7.3	LookupCodes - koodien muut haut	73
5.7.4	IsCodeValid - koodin validointi	76
5.7.5	GetSupportedCodesetServices - koodistossa tuetut palvelut	77
5.7.6	GetCodesetInfo - Koodiston hallinnolliset tiedot	78
5.7.7	ListLanguages – koodiston tukemat kielet	79
5.7.8	GetCodes - yksilöityjen koodien halutun sisällön hakeminen	79
5.7.9	GetSupportedAttributes - koodiston tietokenttien kysely	81
5.7.10	GetHierarchyDepth – koodistohierarkian syvyys	82
5.8	Code-rajapinta	82
5.8.1	GetDesignation - koodin arvon hakeminen	83
5.8.2	GetParent – ylemmän tason koodi	83
5.8.3	GetStatus - koodin tila	84
5.8.4	GetLocal - koodin paikallisuus	85
5.8.5	LookupCompleteCodedConcept - koodin koko sisällön hakeminen	86
5.8.6	LookupProperties - koodin halutun sisällön hakeminen	87
5.8.7	GetHierarchyLevel - koodin hierarkiataso	88
5.9	Virhetilanteet	88
5.10	Laajennukset ja jatkokehitys	89
5.11	Toteutuksiin kohdistuvia vaatimuksia	90
5.11.1	Toteutuskohdista ratkaistavat seikat	90
5.11.2	Määrittelyn mukaisuudesta varmistuminen	90
6	SISÄLTÖMÄÄRITYKSET: RAJAPINNAN LIITTÄMINEN ERI KOODISTOIHIIN	92
6.1	ICD-10 sisältö-määrittely	92
7	TOTEUTUKSET JA JATKOKEHITYS	95
8	LÄHTEET	96
LIITE 1.	STAKESIN KOODISTOPALVELUN JA HL7 MDF:N TIETOELEMENTIT	99
LIITE 2.	TEKNISEN MÄÄRITYKSEN OPERAATIOLISTAUUS (V2.0)	101
LIITE 3.	MÄÄRITYKSEN TASOJEN VAATIMUKSET TOTEUTUKSILLE	103
LIITE 4.	TEKNISEN MÄÄRITYKSEN XML-ELEMENTIT	105
LIITE 5.	KOMMENTOINTIKIERROKSELLA SAADUT KOMMENTIT JA KYSYMYKSET VASTAUKSINEEN	107

1 JOHDANTO

Eri ohjelmistoissa tiedon merkityksen huomioon ottavan (semanttisen) yhteentoimivuuden edellytyksenä on tiedon sallittujen arvojen (arvojoukkojen) yhtenäisyys tai korrelointi varsinaisen tiedon siirtämisen lisäksi. Yhteentoimivuus edellyttää, että sovellukset voivat jakaa ja käyttää samoja rajattuja arvojoukkoja (mm. koodistoja) nimetyissä tietokentissään. Näitä tarpeita on sekä terveydenhuollon sovellusintegraatioissa yleensä että varsinkin sähköisessä potilaskertomuksessa. Koodistojen epäyhteensopivuus, paikalliset ja sovelluskohtaiset koodistot sekä samojen koodistojen epäyhteensopivat versiot aiheuttavat runsaasti yhteensopivuusongelmia sovellusohjelmistojen välillä terveydenhuollossa ja mm. hankaloittavat yhteismitallisen ja vertailtavan päätöksentekoa tukevan tiedon saantia eri järjestelmistä.

1.1 PlugIT-projekti

PlugIT on vuosina 2001-2004 toteutettu valtakunnallinen Tekes-rahoitteinen tutkimus- ja kehittämishanke, joka tuottaa avoimia ohjelmistorajapintojen määrittämiä sekä niihin liittyviä menetelmiä ja osaamista terveydenhuollon ohjelmistoyrityksille ja niiden asiakkaille. Hankkeen tavoitteena on tukea terveydenhuollon palvelutoimintaa ohjelmistotuotannon palveluketjun kautta, paremmin integroituvien ohjelmistokokonaisuuksien avulla (PlugIT 2004).

Tutkimuksen sekä rajapintojen määrittelyn ovat toteuttaneet neljä Kuopion Centekin yksikköä yhteistyössä 15 ohjelmistoyrityksen, kuuden sairaanhoitopiirin (mukaan lukien kaikki yliopistolliset sairaanhoitopiirit) sekä kahden kunnan tai kuntayhtymän kanssa.

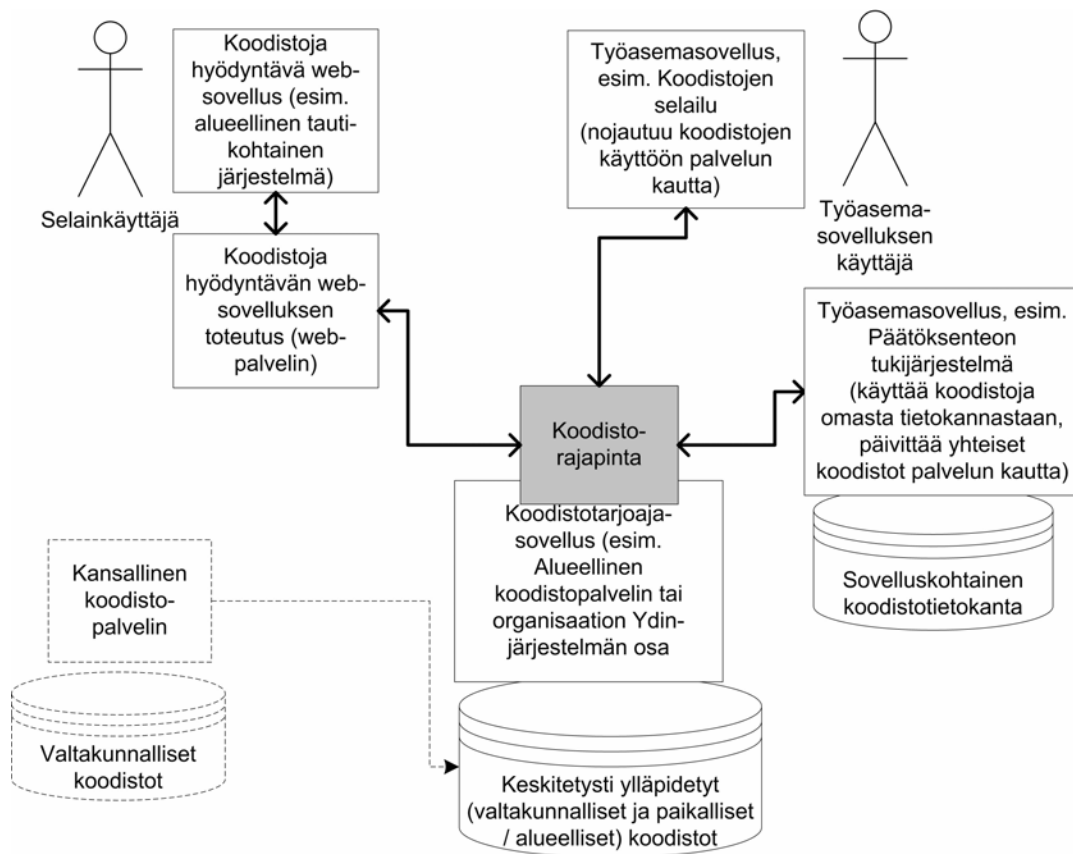
Tämä raportti dokumentoi projektin osana vuosina 2003-2004 tehtyyn koodistojen ohjelmistorajapintojen määrittelytyöhön liittyviä tuloksia. Nämä tulokset on tuotettu lähteessä (Mykkänen ym. 2004c) kuvattuja menetelmiä käyttäen, ja kyseisessä raportissa (luku 7.2) on kuvattu tässä kuvattavien koodistomäärittysten etenemistä, käytettyjä menetelmiä ja saatuja kokemuksia.

1.2 Sovellusten koodistorajapintojen määrittelyt

Eräs PlugIT-projektissa tunnistetuista ja keskeisiksi nähdystä integrointikohteista on ollut tuottaa rajapinnat, joilla sovellukset saavat koodistoihin ja organisaatioihin liittyviä palveluita käyttöönsä esim. potilashallinnon ydinjärjestelmästä tai muusta tällaisista palveluista tarjoavasta sovelluksesta (esim. alueellinen koodistopalvelin, sairaalan paikallinen terminologiapalveluita tarjoava sovellus tai terminologiatietojärjestelmä).

Terveydenhuollon sovellusten ja PlugIT-projektin *ydinrajapintojen* perusajatuksena on se, että joukko sovelluksia tarvitsee ja voi käyttää hyväkseen ydinjärjestelmän tarjoamia ydinpalveluita yhtenäisten rajapintojen kautta. Nämä rajapinnat ovat avoimia ja uudelleenkäytettäviä, ja useat eri ydinjärjestelmät voivat tarjota samoja rajapintoja. Sovellusten kannalta voidaan nojautua siihen, että rajapinnan toteuttava ohjelmisto on saatavilla ja tarjoaa rajapinnassa määritellyt palvelut. Näin näitä palveluita ei tarvitse toteuttaa erikseen esim. jokaiseen erillisjärjestelmään, ja liittyminen palvelun toteuttaviin keskitettyihin sovelluksiin on suoraviivaista ja yhdenmukaista eri paikoissa ja eri sovellusten kohdalla. Tämä tehostaa ja helpottaa järjestelmien käyttöönottoa eri ympäristöissä ja vähentää paikallisen sovitustyön tarvetta toistuvassa integroinnissa. Rajapinnan kautta käytettävät tiedot myös ylläpidetään keskitetysti rajapinnan palveluista tarjoavassa sovelluksessa (ks. kuva 1.1).

Stakesin *Koodistopalvelin* –projektissa määritellään joukko koodistoja, joita tarjotaan keskitetysti valtakunnallisella tasolla. Nämä koodistot siirretään yhdeltä keskitetyltä koodistopalvelimelta koodistojen hyödyntäjille, joita voivat olla esim. alueelliset koodistopalvelimet tai ydinjärjestelmät. Koodistopalvelin-projektissa määritellään myös siirrettävien koodistojen tietosisällöt valituille koodistoille sekä tavat, joilla koodistoja hyödyntävä esim. alueellinen palvelin voi pyytää haluamansa tiedot (esim. kaikki tai tietyn päivämäärän jälkeen muuttuneet koodistojen perustiedot tai koodiarvot). (Koodistopalvelin 2003). Myös HL7 on määritellyt kansainvälisesti mm. koodistoihin liittyviä tietojoukkoja, jotka on syytä vähintään sisällyttää koodistoja koskeviin integrointiratkaisuihin. Stakesin koodistopalvelimen ratkaisu perustuu nykyisellään koodistojen kopiointiin niitä hyödyntävään järjestelmään, ja esim. erillissovellukset eivät saa ottaa suoraan yhteyttä kansalliseen palvelimeen. PlugIT-projektin koodistomäärittelyissä on hyödynnetty Stakesin koodistopalvelun tietosisältöön liittyviä määrittelyjä ja määritellyt rajapinnat, joiden avulla sovellukset saavat tarkempia ja sovellusläheisempiä palveluita (esim. ydinjärjestelmältä tai alueelliselta koodistopalvelulta). Näiden rajapintojen toteuttaminen kansallista palvelua hyödyntäviin ohjelmistoihin helpottaisi yhteisesti ja valtakunnallisesti saatavilla olevien koodistojen ”arkikäyttöön” leviämistä sovelluksissa (ks. kuva 1.1).



Kuva 1.1. Esimerkkejä koodisto-ydin-rajapinnan hyödyntämisestä.

PlugIT-projektissa on aiemmin kerätty integrointiin liittyviä vaatimuksia myös koodistojen hyödyntämistä koskien. Nämä vaatimukset sekä myös jotkin kansallisen Terveystietokeskuksen valtakunnallisen sähköisen sairauskertomuksen osahankkeessa esitetyt terminologiajärjestelmiin kohdistuvat vaatimukset ovat monilta osin OMG:n määrittelemän TQS (Terminology Query Services)-standardin vaatimusten mukaisia. TQS-standardin määritellyjä palveluita käytetään koodistopalvelimen tietosi-

sällön lisäksi mallina ratkaisun määrittelyssä, mutta suora standardinmukainen toteutus ei ole järkevää, koska standardissa käytetty integrointiteknologia ei ole levinnyt Suomen terveydenhuollon tietojärjestelmissä ja sen käyttöönotto aiheuttaisi liikaa ylimääräistä työtä osallistuviin sovelluksiin. (ks. luku 2.2). Tarkempaa tietoa määrittelyyn vaikuttaneista taustaselvityksistä ja standardeista on luvussa 2.

Tämän dokumentin rakenne, sisällön osat ja sisältö on tuotettu PlugIT-projektin integrointi-prosessia ja prosessin vaihedokumentteja soveltaen (Mykkänen ym. 2004c). Tähän dokumenttiin on koostettu eri aikoina syntyneitä vaihedokumentteja yhdeksi ”paketiksi”, mutta eri osiot ovat syntyneet vähittäin työn edetessä, ja samoin niiden itsenäinen jatkokehitys on mahdollista. Työstettävät vaatimus- ja liittymämäärittelyt on hyväksytty ja julkistettu PlugIT-projektin hyväksymiskäytännön mukaisesti (Mykkänen ym. 2004c, luku 2.2.2).

Monista keskeisistä koodistoista löytyy lisätietoja ja linkkejä myös raportista ”Standardien arviointi ja valinta terveydenhuollon sovellusintegraatiossa” (Mykkänen ym. 2004b), ja Suomessa hyödynnettäviä koodistoja on lueteltu runsaasti raportissa (Ensio, Ruotsalainen 2004), jossa on nähty myös tarve koodistopalvelun API-rajapintojen kansalliselle suositukselle.

1.3 Dokumentissa käytettyjä käsitteitä

hallinnolliset tiedot	koodistoon liittyvät tiedot mm. koodiston ylläpitäjästä, versiosta, voimassaolosta, ohjelmiston tarjoajasta jne.
Integrointivaatimukset kenttä	dokumentti, jossa määritellään integrointimäärittelytyön tavoitteet koodiston koodin sisältämä yksittäinen tietoelementti, esim. lyhenne, pitkä selite, englanninkielinen selite tai hierarkiataso
koodi(elementti)	koodiston sisältämä peruselementti, josta koodisto koostuu, elementissä on yleensä ainakin koodiarvo ja selite, mutta usein myös muita kenttiä kuten eri kielisiä tai lyhyitä ja pitkiä selitteitä jne.
koodiarvo	koodin yksikäsitteinen, yleensä koodiston sisällä yksilöivä arvo, koodin tunnistee
koodisto	määritelty joukko arvoja, jonka tarkoituksena on kuvata mahdolliset vaihtoehdot tietystä arvojoukosta, koodiston yksittäisissä elementeissä on tyyppillisesti ainakin koodiarvo ja selite, ks. nimikkeistö, luokitus
koodistohyödyntäjä	koodistopalvelua käyttävä sovellus tai komponentti
koodistotarjoaja	koodistopalvelua tarjoava sovellus tai komponentti
luokitus	nimikkeistö, jossa termit on ryhmitelty; jos niitä voidaan jakaa ylä- ja alaluokkiin, kyseessä on hierarkkinen luokitus, ks. nimikkeistö, koodisto
nimikkeistö	määritelty lista tietyn aihepiirin termejä, jotka on muodostettu sovituin säännöin, nimikkeillä ei välttämättä ole koodattuja arvoja
OID	Object IDentifier, numeroista muodostettu merkkijono, jota käytetään oliotunnuksena erilaisissa protokollissa. Merkkijonot määritellään hierarkkisesti ja hierarkkiasot erotetaan toisistaan pisteillä. Myös CEN-tietotyyppimäärittelyn (CEN 2003) mukainen OID-tietotyyppi.
selite	koodiin liittyvä näytettävä selväkielinen arvo
sovellus	mikä tahansa terveydenhuolto-organisaation tietojärjestelmäkokonaisuuden osa, joka on tarkoitettu tietyn erityisen osatoiminnan tukemiseen. Voi olla erillinen kliininen tai potilashallinnollinen sovellus tai osa laajempaa kokonaisjärjestelmää, kertomusjärjestelmä, portaali, aluetietojärjestelmä, yms.

Tekniikkariippumaton liittymämäärittely	dokumentti, jossa määritellään integrointiratkaisun sisältö (mm. toiminnallisuus ja tietosisältö) tavalla, joka voidaan toteuttaa usealla eri tekniikalla
tekninen liittymämäärittely	dokumentti, jossa määritellään integrointiratkaisu valitulla teknisellä toteutustavalla; ratkaisu voidaan näillä valituilla tekniikoilla toteuttaa useisiin eri sovelluksiin
toimiyksikkö	<i>toimintayksikkö</i> , joka on organisaatioyksikkö tai sen osa, joka on tehtäviensä hoitamisessa hallinnollisesti ja taloudellisesti itsenäinen. Sosiaali- ja terveydenhuollon toimintayksiköitä ovat esimerkiksi julkiset ja yksityiset sairaalat sekä niiden osat, terveyskeskukset. Tietojärjestelmien kannalta toimintayksikkö on tietojärjestelmää käyttävä organisaatio tai sen osa, jolla on vaikutusta tietojärjestelmän toimintaan. Organisaatiotiedot muodostavat tietojärjestelmissä erilaisia hierarkioita (Stakes 2002).
TQS	OMG:n <i>Terminology Query Service</i> . Tämän spesifikaation määrittelemät rajapinnat sisältävät yleisiä metodeja, joiden avulla voidaan käyttää terveydenhuollon sovelluksissa tarvittavia luokituksia ja terminologiapalveluita.
Ydinjärjestelmä	terveydenhuollon organisaation (sairaalan tai terveyskeskuksen tms.) tietojärjestelmän ne osat, joissa ylläpidetään keskitetysti tärkeimpiä yhteisiä tietoja, kuten potilastietoja, käyttäjätietoja, koodistotietoja, yms. Voi olla yksi tai useampia erillisiä ydinkomponentteja tai -sovelluksia tai osa kokonaisjärjestelmää (esim. monoliittista perinnejärjestelmää).

1.4 Raportin rakenne

Tämän dokumentin luvussa 2 esitetään taustaselvityksiä (tarvekartoitukset, standardit ja ulkoiset määrittelyt), jotka ovat vaikuttaneet raportin myöhempien lukujen tuotoksiin. Näin ollen luku 2 ei ole välttämätön niille lukijoille, joille nämä taustaseikat eivät ole olennaisia.

Luvussa 3 esitetään tehdyn määrittelytyön tavoitteet, päärajaukset ja ohjelmistotyypit, joihin tehdyt ratkaisut liittyvät. Luku sisältää myös luettelon toiminnoista ja tietosisällöstä, joita tuotetavalta ratkaisulta vaaditaan.

Luvussa 4 kuvataan yleinen ratkaisu, jolla asetetut vaatimukset saavutetaan. Luvussa kuvataan integrointitapa sekä sovellusten vuorovaikutus sekä tunnistettujen rajapintojen toiminnot ja tietosisältö siten, että rajapinnat voidaan tarkentaa haluttaessa eri liittymätekniikoilla. Luvussa 4 kuvataan myös määrittelyksen eri ”tasot”, eli tiettyihin vaatimuksiin liittyvät toimintojoukot.

Luvut 3 ja 4 perustuvat PlugIT-hankkeen dokumenttiin ”Koodistorajapinnat: Integrointivaatimukset ja Tekniikkariippumattomat liittymämäärittelyt, versio 1, 11.11.2003”.

Luvussa 5 on tarkennettu luvun 4 rajapintoja valitsemalla integroinnissa käytettävät rajapintatekniikat. Luku sisältää tarkat tekniset määrittelyt ohjelmistorajapintojen toteuttamiseksi sekä käytetyt XML-merkkaukset ja rakenteet. Lisäksi luvussa kuvataan kuinka eri koodistot liitetään rajapinnan kautta käytettäväksi sisältömäärittelysten avulla. Dokumentin liitteet täydentävät pääosin lukua 5.

Lukuun 6 on koottu vaatimukset, jotka kohdistuvat määrittelytoteuttaviin ohjelmistoihin ja tietojen määrittelyjen mukaisuudesta ja sen toteutamisesta.

Tämän dokumentin luvut 1-4 ja 7 sekä liite 1 ovat taustamateriaalia ja taustatietoa koodistorajapinnoista. Dokumentin luvut 5 ja 6 sekä liitteet 2-5 muodostavat normatiivisen teknisen rajapintastandardin, jota noudattaen voidaan esim. toteuttaa hyväksytyn määrittelyksen mukaisia teknisiä rajapintoja tai tehdä tarjouspyyntöjä.

2 TAUSTASELVITYKSET

Tässä luvussa kuvataan tuotettuihin määrittelyihin liittyviä taustaselvityksiä, standardeja ja ulkoisia määrittelyitä. Tässä luvussa erikseen mainittujen kyselyiden ja standardien lisäksi tarpeiden, vaatimusten ja ratkaisujen lähteinä on käytetty mm. Satakunnan Makropilotin koodistotuotoksia (Isokorpi ym. 2001) ja ulkoisia terminologiapalveluiden selvityksiä (Duke, Crawford 2001). Lisäksi PlugIT-hankkeessa oli integrointitarpeen tunnistamisen jälkeen myös koodisto- ja organisaatorajapintoihin oli kerätty vaatimuksia ja kommentteja mm. puolivuositaisissa työpajoissa (koodistoihin liittyen kolmessa työpajassa vuosina 2003-2004), sähköpostikyselyillä terveydenhuollon tietohallinnon edustajille ja projektiin osallistuville yrityksille sekä kahdenkeskisissä keskusteluissa.

2.1 Nykytila ja kehittämiskohteet

Nykyisellään monet yhteiset, eri sovellusten käyttämät koodistot kopioidaan kunkin sovelluksen omaan tietovarastoon. Joissakin tapauksissa koodistojen jakaminen on toteutettu yhteisen tietokannan tai tietokantanäkymän kautta (esim. useissa Musti-pohjaisissa sekä muutamissa uusissa järjestelmissä sekä niiden integroinnissa). Tietokantapohjainen integrointi on jotakuinkin yksinkertainen ja tehokas tapa jakaa yhteisiä rekistereitä. Se on kuitenkin riippuvaista usein järjestelmäkohtaisesta tietokantamäärittelystä eikä tarjoa tiedon jakamisen lisäksi edistyneempiä terminologiapalveluita, ja tietojen päivitys kunkin sovelluksen omaan tietokantaan voi vaatia ylimääräistä ylläpitotyötä.

Stakesin koodistopalvelin-projektin ensimmäisessä vaiheessa on valittu joukko toteutettavia koodistoja (ks. luku 2.5) sekä määritelty koodistojen siirtoon tarvittavat sanomat (Koodistopalvelu 2003). Määrittelyissä on käytetty SOAP-tekniikalla paketoituja XML-kuvauskielillä kuvattuja koodistosanomiamia. Myös työ organisaatiokoodiston kehittämiseksi on käynnissä.

Koodistoja tuottavat Suomessa mm. Stakes, Kuntaliitto, Työterveyslaitos, Kela, KTL, Lääkelaitos, Säteilyturvakeskus, Duodecim, HL7 Finland ja sosiaali- ja terveydenhuollon organisaatiot. Osa näiden organisaatioiden tuottamista koodistoista tulee jaeltavaksi koodistopalvelimen kautta (Koodistopalvelin 2003).

Koodistoja käyttävät mm. sosiaali- ja terveydenhuollon organisaatiot, Stakes, Kuntaliitto, tilastokeskus, tutkimuslaitokset sekä ohjelmistotoimittajat ja tietotekniikan käyttöpalvelujen tuottajat (Koodistopalvelin 2003). Tässä määriteltävissä ratkaisuisissa keskitytään ohjelmistojen tekijöiden, tietotekniikan käyttöpalvelujen tuottajien sekä sosiaali- ja terveydenhuollon organisaatioiden (sovellusten käyttäjät ja tietohallinto) koodistotarpeisiin sovellusintegraatiossa.

Eri sovellukset käyttävät yleensä kyseisen organisaation potilashallinnon organisaatio- ja toimipisterekistereitä. Lisäksi sovelluksissa on omia organisaatiokoodistoja. Koodeja on olemassa tulosityksikkötasolle, laitostasolle tai toimipistetasolle. Organisaatiotietoa hyödynnetään pääsääntöisesti potilaan tietoa tallennettaessa (organisaatiotieto tallennetaan lokitietona) ja käyttöoikeuksia määritettäessä (mm. potilaan suostumus ja hoitosuhteen määräytyminen). Organisaatiotietoa tarvitaan myös nyky- ja tavoitetilassa lausuntojen kierrätykseen, konsultointitarpeisiin, lähete-palautetoimintaan ja tilastointiin.

Samoja koodistoja tarvitaan hyvin monenlaisissa sovelluksissa, jotka on voitu toteuttaa pitkän ajan kuluessa käyttäen monia eri tekniikoita ja sovellusarkkitehtuureita. Tuotettavissa integrointiratkaisuissa on otettava tämä heterogeenisuus huomioon suosimalla avoimia tekniikoita ja ratkaisuja, jotka mahdollistavat eri tyyppisten (mm. web ja Windows-) sovellusten osallistumisen ratkaisun mukaiseen integraatioon.

Hallinnollisten taustarekisterien ja koodistojen käyttö ulkoisesta ohjelmistosta ohjelmointirajapinnan kautta nähtiin eräänä kehityskohteena PlugIT-osapuolille tehdyssä ohjelmistotuotannon nykytila ja kehittämiskohteet-kyselyssä (ks. ao. taulukot kyselyn tuloksista) (Porali ym. 2004).

Taulukko 2.1. Ohjelmistotuotannon nykytilakyselyn koodistojen käyttöön liittyviä tuloksia (Porali ym. 2004)

Hallinnollisten taustarekisterien ja koodistojen käyttö ulkoisesta ohjelmistosta ohjelmointirajapinnan kautta, nykytila	Frekvenssi (n=10)	Prosentit
ei koskaan	1	10
harvoin	2	20
joskus	3	30
usein	2	20
lähes aina	1	10
en osaa sanoa	1	10

Hallinnollisten taustarekisterien ja koodistojen käyttö ulkoisesta ohjelmistosta ohjelmointirajapinnan kautta, tavoitetaso	Frekvenssi (n=7)	Validit prosentit	Prosentit
joskus	3	42,9	30
usein	3	42,9	30
en osaa sanoa	1	14,3	10
puuttuvat	3	-	30

	Keskiarvo	Keskihajonta	n
Hallinnollisten taustarekisterien ja koodistojen käyttö ulkoisesta ohjelmistosta ohjelmointirajapinnan kautta, nykytila	2,00	1,23	10
Tavoitetaso	2,50	0,55	7

Samassa kyselyssä kysymykseen ”Toteutettavissa järjestelmäintegraatioissa standardien tai yleisten määritysten pohjalta sovittavia asioita”-kysymyksessä:

- asiakasorganisaation organisaatioyksiköt ja organisaatorakenne –kysymyksen tavoitearvo oli 2,86 (nykytila 1,88)
- tietosisällön eri osissa käytetyt koodistot: tavoitearvo 4,00 (nykytila 3,13).

Vastausten jakaumista voidaan tulkita, että vastaajat näkivät yleisten koodistojen ja organisaatietietojen integroinnissa tarvetta standardointiin ja ohjelmointirajapintojen käyttöön, ja ulkoisten ohjelmointirajapintojen käyttöä koodistojen hyödyntämiseksi haluttiin lisätä.

2.2 OMG Terminology Query Services (TQS)

OMG (Object Management Group)-standardointiorganisaation terveydenhuollon ryhmä (Healthcare Domain Task Force) on määritellyt joukon terveydenhuoltospesifejä sovellusten yhteisten palvelujen rajapintoja (Savolainen 2004), joista koodistoja käsittelee TQS (Terminology Query Services). Tämä luku sisältää TQS-standardista tehdyn selvityksen, jonka tavoitteena on ollut arvioida, etsiä ja osoittaa TQS-standardista sellaisia vaatimuksia, osia ja rajapintamäärittelyitä, joita on hyödynnetty projektissa määriteltävissä koodistorajapinnoissa.

Standardin perustiedot ja sen kattama alue

Standardin perustietojen keräämisessä on hyödynnetty PlugIT-projektin ”Standardien arviointi ja valinta terveydenhuollon sovellusintegraatiossa”-raportissa kuvattua arviointimallia (Mykkänen ym. 2004b), jossa on myös viittaukset arvioinnissa käytettyihin viitemalleihin.

Taulukko 2.2. TQS-standardin arviointitaulukko.

Standardi	Terminology Query Service (TQS), ent. Lexicon Query Service (LQS)
Standardointijärjestö / -perhe	OMG Healthcare Domain Task Force (ent. Corbamed, 2000).
Tutkittu versio	1.0, OMG document corbamed/2000-05-01
Kohdealue	
Tarkoitus	Terveydenhuollon terminologia- ja koodistojärjestelmien oliopohjaiset rajapinnat
Kohdealue	Terveydenhuollon hajautetut sovellukset
Ensisijainen RM-ODP-näkökulma	Computation: määrittää oliopalveluita
Muut RM-ODP-näkökulmat	Liittymät on määritelty OMG IDL:llä, toteutus vaatii ORB-tuotteen (Engineering)
Ensisijainen 7-tasoisin mallin taso	Toiminnalliset liittymät (4) – IDL:llä määriteltyjen operaatioiden nimet ja parametrit.
Vaikutukset muille 7-tasomallin tasoille	Edellyttää CORBA-sovellusinfrastruktuuria (2), käyttää OMG IDL:ää (1), edellyttää puhtaasti sovellettuna hajautettua CORBA-arkkitehtuuria (3), edellyttää sovittimia, jos palveluita halutaan käyttää olemassa olevien sovellusten sisältämien termien tai koodien mediaatioon tai liittää palvelujen hyödyntämistä sovelluksiin (6)
Toiminnot ja tietosisältö	
Tieto- tai käsittemalli	Keskeiset käsitteet ja niiden suhteet on määritelty
Tietosisältö	Liittymien parametrit, voidaan käsitellä erilaisia koodistorakenteita ja eri kenttiä sisältäviä koodistoja, joita ei ole määritelty standardissa.
Tietosisällön kuvaaminen	OMG IDL-kielen avulla, käytetyt tietotyypit määritelty
Tietosisällön merkitys	Parametrien merkitys kuvattu standardissa
Toimintomalli	Liittymien operaatiot
Toimintomallin kuvaaminen	OMG IDL-kielen avulla
Toimintojen merkitys	Operaatioiden merkitys kuvattu standardissa
Koodistot tai luokitukset	(implisiittinen; rakennettu koodistojen ja luokitusten hyödyntämistä ja mediaatiota varten). Sisältää mm. ICD-9, UMLS, HL7 v.2.3-koodisto-esimerkkejä.
Ihmisten (käyttäjien) toiminnan kuvaaminen	Kuvaa joukon käyttöskenaarioita osana palvelun kuvausta

Sovellusarkkitehtuuri	
Osien tunnistaminen	Asiakas ja palvelin (hajautettu palvelu)
Yhteistoiminnan perusmalli	Oliopohjaisuus
Integrointitapa	Yhteinen infrastruktuuri (oliosanomavälitin)
Hajautus	Hajautetut palvelut
Kutsutapa	Pyyntö-vastaus, toiminnon (palvelun) suorittaminen
Tekniikka	
Tiedon esitys	OMG IDL
Toimintojen määrittely	OMG IDL
Kutsutekniikka	Operaatiokutsut määritelty OMG IDL:n avulla, josta tyypillisesti generoidaan runko sekä asiakas- että palvelinohjelmiston toteutukselle
Verkkoliikenne	IIOP (CORBA), (tcp/ip-verkossa), palomuurit ongelmallisia
Suoritukseen tarvittava tekninen infrastruktuuri	ORB-tuote, CORBA-nimipalvelu
Toteutuksessa tarvittavat välineet	IDL-kääntäjät, ohjelmointikielet suoritusinfrastruktuurin lisäksi
Suhde järjestelmän elinkaaren vaiheisiin	
Vaatimukset	Määrittelee joukon koodisto- ja terminologiapalveluiden vaatimuksia, joihin vastaa.
Kohdealueen analyysi	Käyttää perinpohjaista käsitelmää, jolla on kuvattu yksinkertaisten koodistojen lisäksi monimutkaisia terminologioiden elementtien suhteita
Tietosisällön ja toimintojen määrittely	ks. yllä (kohta toiminnot ja tietosisältö)
Tekninen määrittely ja suunnittelu	Sisältää valmiit liittymien määrittelyt IDL:llä
Toteutus	IDL-kieliset liittymät voidaan toteuttaa eri välineillä ja ohjelmointikielillä
Asennus ja käyttöönotto	Vaatii suoritussympäristöön palvelimen, jolle palvelu asennetaan, ja ORB-tuotteen, joka huolehtii verkkokommunikaatiosta asiakkaan ja palvelimen välillä.
Ylläpito ja uudet versiot	Standardi ollut hyväksytty, käytössä ja muuttumaton vuodesta 2000. OMG:n hyväksymisprosessi vaatii toteutuksen tuotteissa ennen hyväksymistä. Standardissa on määritelty useita Conformance Level-tasoja (standardin mukaiset tuotteet voivat toteuttaa vain osan määritellyistä liittymistä)
Arvio levinneisyydestä	
Kansainvälisesti	Ei laajasti käytössä, arvioijien tiedossa olevat toteutukset: 3M Healthcare Data Dictionary (kaupallinen toteutus, 3M Care Innovation Component, kaupallinen terminologiajärjestelmä); PropeR-EHR (sähköisen potilaskertomuksen osa, Dept. Medical Informatics, Maastricht University, Hollanti, tutkimusprojektin tuotos, 2003)
Suomessa	Ei käytössä, eivätkä OMG:n terveydenhuoltospesifit standardit ole levinneet yleiseen käyttöön.
Suhde muihin standardeihin	
Vahva suhde OMG:n yleisiin middleware-tekniikkastandardeihin, OMG:n terveydenhuoltoryhmä neuvottelee yhteistyöstä HL7:n kanssa. Sisällöllisissä esimerkeissä käytetty mm. ICD-9, HL7 ja UMLS-koodistostandardeja.	

Muuta olennaista

Standardi on laaja ja suhteellisen monimutkainen ja kattava joukko terminologiajärjestelmien oliopalveluiden määrittäjiä. Sen täysimittainen toteuttaminen tuotteessa on suuritöistä, joskin standardissa määriteltynä minimi-toteutukseen (conformance level) riittää kahden operaation toteutus. Kaikilla tasoilla standardinmukaisuuden vaatiminen esim. koodistoja tarjoavissa sovelluksissa ei ole järkevää.

OMG-standardien jatkokehitystyö ja mahdollinen siirto muille kuin CORBA-alustoille riippuu OMG-terveydenhuoltoryhmän aktiivisuudesta (viime aikoina ei aktiivista), MDA-määrittelytyön (ks. Mykkänen 2004c) etenemisestä ja siihen liittyvästä OMG:n jäsenyritysten toiminnasta.

TQS-spesifikaatio määrittelee rajapintoja, joiden toteutusten avulla sovellukset voivat käyttää erilaisia terveydenhuollon luokituksia keskitettyjen palveluiden avulla. Palvelut vaihtelevat yksinkertaisista keskitetyistä koodistopalveluista ja valintalistojen generoinnista monimutkaisiin eri terminologioiden välisten suhteiden ja koostettujen, eri sovelluksissa eri tavoin määriteltujen tai puutteellisten käsitteiden hyödyntämiseen.

TQS ei ota kantaa luokitusten sisältöön, ja määritelmää voitaisiin käyttää myös muilla toimialoilla. Koska yksikäsitteisten, täsmällisten luokitusten käyttö on erityisen tärkeää terveydenhuollossa, on Healthcare Domain Task Force ottanut tämän määrittelyn kehitettäväkseen. TQS-määrittelytyön tavoitteeksi asetettiin tuottaa spesifikaatio (OMG 2000, TQS 2000),

- jota voitaisiin käyttää rajapinnan toteuttamiseen mihin tahansa merkittävään terveydenhuollon luokitukseen. Luokitus voi olla yksinkertainen koodilista mutta myös monimutkainen hierarkkinen luokitusjärjestelmä.
- jonka avulla edistettäisiin kliinisten prosessien ja niiden tulosten analysointia yhtenäisten ja täsmällisten luokitusten ansiosta.
- jonka avulla edistettäisiin yhtenäistä tiedon esitystapaa edellyttävien päätöksentekoa tukevien välineiden käyttöä.
- jonka avulla voitaisiin edistää tiedonsiirtoa eri organisaatioiden välillä.

TQS-spesifikaatiossa (TQS 2000) on määritelty lista tyypillisimmistä käyttötapauksista, jotka on jaoteltu kuuteen ryhmään. Lista ei ole kattava, mutta siihen on pyritty saamaan kaikki yleisimmät tapaukset, joissa TQS:n määrittelemistä palveluista olisi hyötyä:

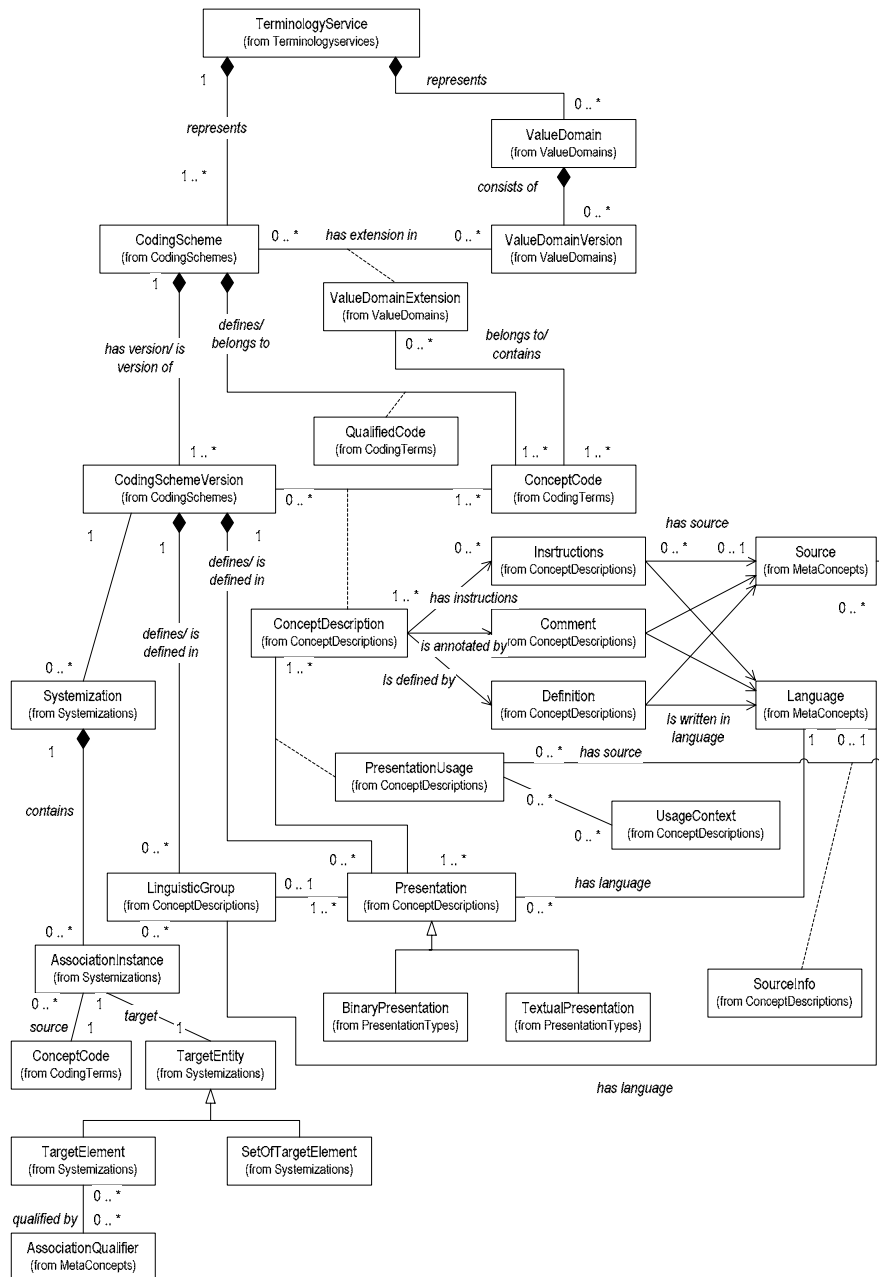
- Tiedon hakeminen koodistosta:
 - Käsitteen hakeminen sovellukseen tietyllä koodilla.
 - Koodin hakeminen tarkan tai osittain annetun käsitteen avulla
 - Koodin hakeminen avainsanoilla.
 - Tarkimman mahdollisen käsitettä vastaavan koodin hakeminen vaiheittain
 - Tiettyyn sovelluksen kenttään mahdollisten koodien ja vastaavien käsitteiden hakeminen
 - Kentän validointi. Sovelluksen kentässä olevan koodin oikeellisuuden tarkistaminen.
 - Valintalistan luominen
- Tietojen esittäminen. Terminologiapalvelujen käyttäminen koodattujen data-elementtien muuttamiseksi tiettyyn (ihmisen tai koneenluettavaan) muotoon, esim. koodin määrittelemän käsitteen esittäminen sovelluksessa annetuilla ehdoilla
- Välitys (mediation). Terminologiapalvelujen käyttäminen eri koodistoista haettujen tietojen käsittelyyn.
- päätöksentekoa tukevien välineiden käyttö
- Selaaminen (browsing). Terminologiapalvelujen käyttäminen terminologia-järjestelmän rakenteen selvittämiseen.

- Yhdistelmäkäsitteiden (Composite Concept) käsittely. Terminologiapalvelujen käyttäminen yhdistelmäkäsitteiden validointia, muuttamista ja yksinkertaistamista.

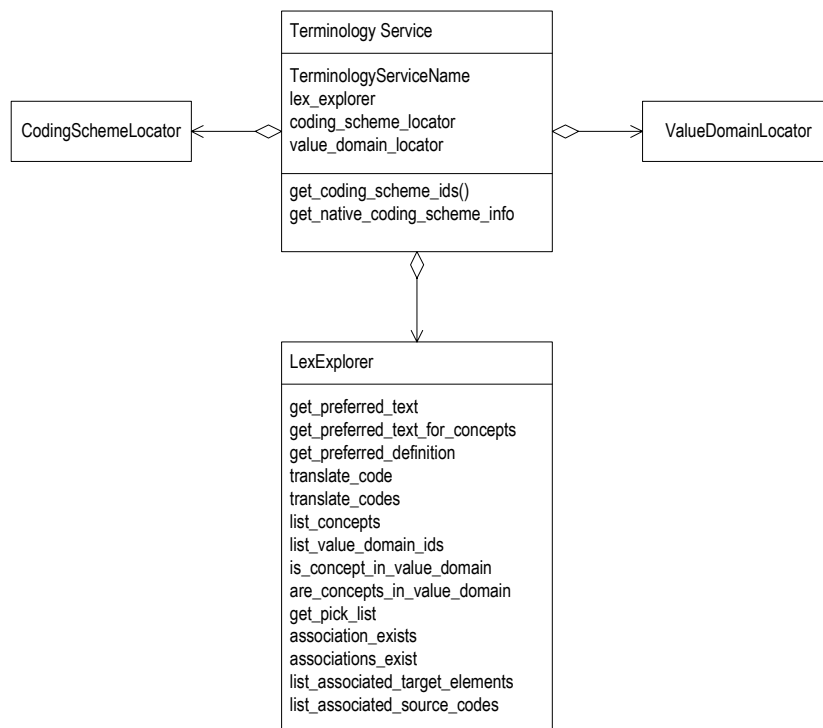
Tässä luvussa käsitellään tarkemmin osaa yllä kuvatuista tapauksista ja niihin liittyvistä operaatioista, jotka ovat koodistorajapintojen kannalta kiinnostavia. Luvun 2.2.2 tietomalli ja rajapinnat ovat syventävää perustietoa standardista.

Tietomalli ja rajapinnat

Kuvassa 2.1 on esitetty yleisellä tasolla TQS:n viitemalli, jossa on määritelty entiteetit, joita käytetään TQS:n IDL-kuvauksessa. Malli on kuvattu TQS-määrittelyssä yksityiskohtaisesti. Terminologiajärjestelmän ei tarvitse sisältää kaikkia mallin entiteettejä ollakseen TQS-standardin mukainen.



Kuva 2.1: TQS:n Viitemalli (Reference Model) (TQS 2000).



Kuva 2.2. TerminologyServices-moduulin TerminologyService-, LexExplorer- ja ValueDomainLocator- rajapinnat (TQS 2000):n tietojen perusteella.

Kuvassa 2.2 on esitetty TerminologyServices palvelun rajapinnat TerminologyService, LexExplorer, CodingSchemeLocator ja ValueDomainLocator. TQS:n minimitoteutus sisältää TerminologyService ja LexExplorer-rajapintojen toteutuksen.

LexExplorer rajapinnan metodien avulla voidaan toteuttaa yksinkertaisia yleisesti tarvittavia terminologiajärjestelmäpalveluja. Laajemmissa toteutuksissa voidaan LexExploreriin lisäksi toteuttaa LexExploreria monipuolisemmat rajapinnat ValueDomainVersion ja CodingSchemeVersion, jossa on määritelty mm. metodi, jolla voidaan hakea annettua tekstiä vastaavaa koodi. Näiden rajapintojen toteutuksia käsitellään CodingSchemeLocator ja ValueDomainLocator-rajapintojen kautta.

LexExplorerin metodit muut paitsi `get_preferred_text` ja `get_preferred_text_for_concepts` voivat aiheuttaa poikkeuksen tai palauttaa null-arvon. `Get_preferred_text` metodi palauttaa koodiin (QualifiedCode) liittyvän tekstin ja `get_preferred_text_for_concepts` palauttaa tekstit useammalle koodille.

Käyttötapaukset, rajapinnat ja metodit

Tässä luvussa on otettu tarkempaan tarkasteluun aiemmin mainittuja käyttötapauksia ja etsitty standardista näihin käyttötapauksiin ja vaatimuksiin liittyviä rajapintoja ja operaatioita. On huomattava, että tässä käsitellyt operaatiot ja tiedot ovat vain pieni osajoukko standardissa määritellyistä.

Taulukko 2.3. TQS:n käyttötapauksia ja operaatioita.

Käyttötapaus	Rajapinta:Metodi	In	Out/ Metodin palauttaman arvon tyyppi
Käsitteen hakeminen sovelukseen tietyllä koodilla	LexExplorer: get_preferred_text	QualifiedCode a_qualified_code tiettyyn koodistoon (Coding-Scheme) liitetty ConceptCode UsageContextIdSeq context_ids UsageContextId-sekvenssi UsageContextId =koodin esitysmuodon (presentation) määrittelevän käyttökontekstin (esim. tietty sovellus) tunniste	Metodin palauttaman arvon tyyppi: IntlString (=string)
	CodingSchemeVersion: get_preferred_text	ConceptCode a_code	Metodin palauttaman arvon tyyppi: IntlString
	CodingSchemeVersion: get_text_for_context	ConceptCode a_code UsageContextIdSeq context_ids	Metodin palauttaman arvon tyyppi: IntlString
Koodin hakeminen tarkan tai osittain annetun käsitteen avulla	CodingSchemeVersion: get_concepts_by_text (tarkka käsite)	String text	Metodin palauttaman arvon tyyppi: ConceptCodeSeq (koodisekvenssi)
	CodingSchemeVersion: match_concepts_by_string	IntlString match_string haettava teksti unsigned long how_many palautettava maksimimäärä	WeightedResultSeq weighted_results WeightedResult-sekvenssi. WeightedResult= ConceptInfo+hakua vastava teksti (matching_text)+ tuloksen vastaavuutta "match_string":iin kuvaava paino (weight) (luku välillä 0.0 ...0.1) WeightedResultsIter weighted_result_iter WeightedResults-iteraattori. WeightedResultsIter on rajapinta, joka kuten muut iteraattorirajapinnat, määrittelee mm. metodin, jolla voidaan hakea (korkeintaan) kutsussa määritellyn kokoisia WeightedResult-sekvenssejä.
Koodin hakeminen avainsanoilla	CodingSchemeVersion: match_concepts_by_keywords	OrderedIntlStringSeq keywords unsigned long how_many	WeightedResultSeq weighted_results WeightedResultsIter weighted_results_iter

Tarkimman mahdollisen käsitettä vastaavan koodin hakeminen vaiheittain	LexExplorer: list_associated_target_elements voidaan hakea koodiin AssociationID:n määrittelemällä tavalla liittyviä muita koodeja	QualifiedCode qualified_code koodi AssociationId association_id assosiaation tyyppi boolean direct_only haetaanko vain suoraan ko. koodiin liittyviä koodeja unsigned long how_many palautettava maksimimäärä	TargetElementSeqSeq related_target_seq TargetElementSeqIter related_target_iter
Koodiston tai koodi-valikoiman koodien ja vastaavien käsitteiden listaus	LexExplorer: list_concepts	CodingSchemeId coding_scheme_id koodisto unsigned long how_many palautettava maksimimäärä	ConceptInfoSeq concept_info_seq ConceptInfo-sekvenssi ConceptInfo = ConceptCode + vastaava teksti ConceptInfoIter concept_info_iter ConceptInfo -iteraattori
	CodingSchemeVersion: list_concepts	unsigned long how_many	ConceptInfoSeq concept_info_seq ConceptInfoIter concept_info_iter
Kentän validointi	CodingSchemeVersion: is_valid_concept	ConceptCode a_code	Metodin palauttaman arvon tyyppi: boolean
Valintalistan (pick list) luominen	LexExplorer: get_pick_list	ValueDomainId value_domain_id UsageContextIdSeq context_ids	PickListSeq pick_list PickListIter pick_list_iter
	ValueDomainVersion: get_pick_list	UsageContextIdSeq context_ids	PickListSeq pick_list PickListEntry-sekvenssi PickListEntry = QualifiedCode + vastaava teksti + Boolean-arvoinen is_default PickListIter pick_list_iter PickList-iteraattori.
	ValueDomainVersion: get_pick_list_for_scheme	CodingSchemeId coding_scheme_id UsageContextIdSeq usage_context_ids	PickListSeq pick_list PickListIter pick_list_iter

LexExplorer-rajapinnan metodien avulla voidaan toteuttaa yksinkertaisia yleisesti tarvittavia terminologiajärjestelmäpalveluja. Laajemmissa toteutuksissa voidaan LexExploreriin lisäksi toteuttaa LexExploreria monipuolisemmat rajapinnat ValueDomainVersion ja CodingSchemeVersion, jossa on määritelty mm. metodi, jolla voidaan hakea annettua tekstiä vastaavaa koodi. Näiden rajapintojen toteutuksia käsitellään CodingSchemeLocator ja ValueDomainLocator-rajapintojen kautta.

Toteutukset ja vastaavuudet (conformance)

TQS-standardissa määritellään minimitason lisäksi vastaavuusluokat CodingSchemeLocator, CodingSchemeVersion, PresentationAccess, LinguisticGroupAccess, SystemizationAccess, Systemization, ValueDomainLocator ja ValueDomainVersion. Luokkien vaatimat rajapinnat on esitetty taulukossa 2.4. Taulukon soluissa ainakin yksi rajapinta "o":lla merkityistä (optional) on toteutettava, jotta rajapinta olisi kyseisen vastaavuustason mukainen, ja tähdellä merkityt rajapinnat ovat tasolla pakollisia.

Taulukko 2.4. TQS:n vastaavuusluokat.

Conformance Class	Terminology Service	Lex Explorer	CodingSchemeLocator	CodingSchemeVersion	CodingSchemeVersionAccess	Presentation Access	Linguistic GroupAccess	Systemization Access	Systemization	ValueDomain Locator	ValueDomain Version	Advanced Query-Access
Minimitaso	*	*										
CodingSchemeLocator	*		*		o	o	o	o				o
CodingSchemeVersion	*			*								
PresentationAccess	*					*						
LinguisticGroupAccess	*						*					
SystemizationAccess	*							*	*			
Systemization	*								*			
ValueDomainLocator	*									*		
ValueDomainVersion	*										*	

TQS-standardin mukaisia toteutuksia ohjelmistoissa ei ole löydetty runsaasti. Taulukossa 2.5 on lueteltu tutkimusryhmän löytämien TQS-toteutusten tietoja. 3M Healthcare Data Dictionary –terminologiajärjestelmä hyödyntää TQS-komponenttia. Järjestelmä sisältää useita keskeisiä koodistoja mm. LOINC, UMLS ja ICD9-CM ja käsitteiden kokonaismäärä on yli 800 000. Data Dictionary sisältää myös tietämyskannan (knowledge base) jonka avulla voidaan kuvata mm. käsitteiden hierarkkisia suhteita koodistossa. (3M 2002).

Taulukko 2.6. TQS-toteutuksia (van der Linden ym. 2003, 3M 2002).

Toimittaja	Tuotenimi	Tyyppi	vastaa- vuustasot
Dept. Medical Informatics, Maastricht University, PropeR-projekti	PropeR EHR, TQS-toteutus sähköiseen potilaskertomuksen osa.	Tutkimus-projektin tuotos	-
3M Health Information Systems	3M Healthcare Data Dictionary, hyödynnetään TQS-toteutusta (3M Care Innovation Component)	Kaupallinen toteutus	-

Yhteenveto

TQS-standardin ensisijaiset määrittelyt, eli liittymämääritykset toiminnallisella tasolla (sisältäen operaatioiden ja niiden parametrien kuvaukset) näyttävät vastaavan monia vaatimuksia, joita koodistorajapinnoille ja koodistotietojen hyödyntämiselle on asetettu PlugIT-projektissa ja myös kansallisessa Terveystieteen sähköisen sairauskertomuksen osahankkeessa. Standardin operaatioiden joukossa on valmiiksi määriteltyjä rajapintoja, jotka edesauttavat yhteisten koodistojen käyttöä,

sovellusten liittämistä koodistopalveluihin ja jatkossa mahdollisesti myös päätöksenteon tukijärjestelmien tarvitsemia koodistopalveluita (joita ei käsitellä tarkemmin tässä yhteydessä).

Standardin käyttöönotto sellaisenaan suomalaisissa terveydenhuollon sovelluksissa ei kuitenkaan ole järkevää, etenkin koska CORBA ja IDL-tekniikat, joihin standardi nojautuu, sekä niiden vaatima tekninen infrastruktuuri eivät ole levinneet riittävän laajaan käyttöön Suomen terveydenhuollon sovelluksissa, ja integrointiratkaisuissa on huomioitava myös olemassa olevien sovellusten teknisen liittämisen helppous (Mykkänen ym. 2004c).

Onkin järkevää hyödyntää standardissa määriteltäviä palveluita toiminnallisena mallina määriteltäville rajapinnoille siltä osin kuin standardissa määritellyt vaatimukset vastaavat Suomessa esiin tuotuja vaatimuksia, ja määritellä tekniset ratkaisut suomalaisten vaatimusten ja integroinnissa Suomessa hyödynnettävien tekniikoiden mukaisesti. Standardin tällainen hyödyntäminen on mahdollista esim. PlugIT-projektin integrointituotosten Integrointivaatimukset ja Tekniikkariippumattomat liittymämäärittelyt-tasoilla (Mykkänen ym. 2004c). Tällainen hyödyntäminen mahdollistaa tuotteiden jatkokehityksen siten, että samat palvelut voidaan helpommin toteuttaa myös CORBA- (tai joillakin muilla) tekniikoilla, jos niitä viedään markkinoille, joilla nämä tekniikat ovat käytössä. Näin myös säästytään ”pyörän uudelleen keksimiseltä”, ja laajalta vaatimusten keräys- ja tarkennuskierrokselta niiltä osin kuin valmis standardi jo vastaa suomalaisia vaatimuksia.

2.3 HL7 version 3 koodistojen tietotyypit

HL7-viesteissä ja niitä käsittelevissä ohjelmistoissa on koodatulle tiedolle ja yksilöinnille määriteltä useita tietotyypppejä. Eri tietotyypeillä voidaan esittää sekä yksinkertaisia koodattuja arvoja että monipuolisia terminologiajärjestelmien välisiä käsitteitä. HL7 version 3 koodattuun tietoon ja yksilöintiin liittyviä tietotyypppejä ovat (HL7 2002):

- CS (Coded Simple Value): Koodattu data. Sisältää koodiarvon ja sen selityksen. Koodisto on kiinnitetty HL7-standardissa. Koodisto ja arvot on kiinnitetty.
- CV (Coded Value): Koodattu data. Sisältää koodiarvon lisäksi koodin selityksen ja koodiston ja mahdollisesti alkuperäisen tekstin (josta koodi on aikaansaatu).
- CE (Coded With Equivalents): Koodattu data. Kuten CV, mutta koodi voidaan ilmoittaa myös muiden kuin peruskoodiston avulla. Koodistoa ei ole kiinnitetty. Koodatun tiedon arvo voidaan esittää usealla eri koodistolla.
- CD (Concept Descriptor): Koodattu data. Kuten CE, mutta kohteena oleva käsite voidaan ilmoittaa useiden eri ristikkäisten koodistojen ”leikkauksena”, esim. vasen jalka.
- II (Instance Identifier): Objektin tunniste. Perustuu ISO:n OID:n /object identifier) käyttöön (ks. luku 2.6).

PlugIT-koodistorajapintojen tietosisällön määrittelyssä on hyödynnetty HL7 Message Development Framework-dokumentin (MDF 1999) mukaista koodistojen tietosisältöä (ks. liite 1).

2.4 HL7 Common Terminology Services (CTS)

HL7 Version 3 integrointimalli perustuu vahvasti yhteisesti määriteltävien koodistojen ja terminologioiden käyttöön (ks. Mykkänen ym. 2004c). Tällaista käyttöä tukemaan HL7:ssä on määriteltä HL7 Common Terminology Services (HL7 CTS) –rajapinnat, jotka ovat (samoin kuin PlugIT-hankkeessa määriteltävät rajapinnat) API-tyyppisiä ohjelmistorajapintoja. CTS-rajapinnat on tarkoitettu HL7 version 3 ohjelmistoille terminologiasisältöjen saantiin (Solbrig 2003).

CTS-palveluita ei ole tarkoitettu täydelliseksi terminologiapalveluiksi kuten TQS-rajapintoja. Palveluiden kattama alue on rajoitettu toiminnallisuuteen, jota tarvitaan HL7 version 3

mukaisissa ohjelmistoissa. CTS:n tavoitteena ei myöskään ole olla yleiskäyttöinen kyselykieli, eikä määrittäminen ota kantaa siihen, kuinka palvelut toteutetaan tai esim. kuinka palvelun toteutus löydetään, vaan palvelussa oletetaan että kulloissakin toteutuksissa nämä seikat hoidetaan sopivalla tavalla.

CTS määrittelee kaksi kerrosta HL7 versio 3-viestejä käsittelevien sovellusten ja sanastojen väliin. Viestisovellusten käyttämä ylempi kerros, *Message API* on yhteydessä viestejä käsitteleviin ohjelmistoihin sanastoalueiden (vocabulary domains), käyttökontekstien, arvojoukkojen, koodattujen tietoarvojen ja muiden HL7-viestimallin ominaisuuksien avulla. Alempi taso, *Vocabulary API*, tarjoaa terminologiapalveluita koodistojen, koodien, selitteiden, suhteiden ja muiden koodisto- ja terminologiakäsitteiden avulla.

Message API on HL7-spesifi. Sen päätarkoitus on mahdollistaa monentyyppisten viestien käsittelysovellusten luoda, validoida ja muuntaa CD-tietotyyppipohjaista tietoa yhdenmukaisella ja toistettavalla tavalla.

Vocabulary API on tarkoitettu yleiskäyttöiseksi. Sen avulla sovellukset voivat hakea tietoa eri koodistoista ja terminologioista yhdenmukaisesti ja tarkasti määriteltujen rajapintojen avulla. Message API hyödyntää Vocabulary API:a. Esimerkiksi viestejä käsittelevä sovellus voi kutsua Message API:n fillInDetails-operaatiota täyttääkseen viestiin tietyn koodin halutun kielisen selitteen, jolloin Message API kutsuu edelleen Vocabulary API-rajapinnan operaatioita hakeakseen selitteitä, koodistojen nimiä ja versioita jne.

Taulukossa 2.7 on lyhyesti arvioitu CTS-määrittystä käyttäen samaa ”Standardien arviointi ja valinta”-kaavaa (Mykkänen ym. 2004b) kuin TQS:n arvioinnissa.

Taulukko 2.7. CTS-standardin arviointitaulukko.

Standardi	HL7 Common Terminology Services (CTS) (Solbrig 2003)
Standardointijärjestö / -perhe	HL7, HL7 version 3
Tutkittu versio	2003, versiohistoria jätetty tarkoituksella pois määrittämisestä maininnalla ”päivitetään tarvittaessa”
Kohdealue	
Tarkoitus	Terveystieteiden terminologia- ja koodistojärjestelmien rajapinnat
Kohdealue	Terveystieteiden sovellukset, erityisesti HL7 v3 viestien käsittely (Message API), yleiskäyttöiset terminologiapalvelut (Vocabulary API) ja koodistojen vastaavuudet (Translation).
Ensisijainen RM-ODP-näkökulma	Computation: määrittää palvelurajapintoja ja operaatioita
Muut RM-ODP-näkökulmat	Information: Allaoleva tietomalli määritelty. Engineering: Liittymät on määritelty OMG IDL:llä, suora IDL-toteutus vaatisi ORB-tuotteen
Ensisijainen 7-tasoisien mallien taso	Toiminnalliset liittymät (4) – IDL:llä määriteltävien operaatioiden nimet, parametrit ja merkitykset.
Vaikutukset muille 7-tasomallien tasoille	Suora toteutus edellyttää CORBA-sovellusinfrastruktuuria tai muunnosta toisille tekniikoille (2), käyttää OMG IDL:ää (1), edellyttää puhtaasti sovellettuna hajautettua CORBA-arkkitehtuuria (3), hyödyntää HL7 version 3 viestimallia ja koodistojen metatietoja (6)
Toiminnot ja tietosisältö	
Tieto- tai käsittemalli	Keskeiset käsitteet ja niiden suhteet on määritelty (erikseen Message API, Vocabulary API ja Translation Model)
Tietosisältö	Liittymien parametrit, voidaan käsitellä erilaisia koodistorakenteita ja eri kenttiä sisältäviä koodistoja, joita ei ole määritelty standardissa.
Tietosisällön kuvaaminen	OMG IDL-kielen avulla, käytetyt tietotyypit määritelty, struct-tietorakennemäärittelyä

Tietosisällön merkitys	Parametrien merkitys kuvattu standardissa
Toimintomalli	Liittymien operaatiot
Toimintomallin kuvaaminen	OMG IDL-kielen avulla
Toimintojen merkitys	Operaatioiden merkitys kuvattu standardissa
Koodistot tai luokitukset	(implisiittinen; rakennettu koodistojen ja luokitusten hyödyntämistä ja mediaatiota varten). Määrittelyssä käytetyt HL7- ja ISO-koodistot OID-tunnisteineen lueteltu
Ihmisten (käyttäjien) toiminnan kuvaaminen	Liitteessä kuvattu vaatimuksia lähinnä sovellusten rakentajan ja sovellusten näkökulmasta
Sovellusarkkitehtuuri	
Osien tunnistaminen	Kahden tason rajapinnat Message Processing Application:in ja Vocabulary:n välissä, tarkennus jätetty avoimeksi (IDL tyypillisesti käytössä hajautetuissa rajapinnoissa)
Yhteistoiminnan perusmalli	Palvelupohjaisuus, tilattomat palvelut (yksilöintitieto parametreina useimmissa kutsuissa)
Integrointitapa	jätetty avoimeksi, suora IDL-käyttö edellyttää oliosanomavälitinten käyttöä
Hajautus	jätetty avoimeksi
Kutsutapa	Pyyntö-vastaus, toiminnon (palvelun) suorittaminen
Tekniikka	
Tiedon esitys	OMG IDL, struct-tietorakenteet
Toimintojen määrittely	OMG IDL-operaatiomäärittelyt
Kutsutekniikka	Operaatiokutsut määriteltyt OMG IDL:n avulla, josta tyypillisesti generoidaan runko sekä asiakas- että palvelinohjelmiston toteutukselle
Verkkoliikenne	(jos käytetään suoraan IDL:ää): IIOP (CORBA), (tcp/ip-verkossa), palomuurit ongelmallisia, määrittelyssä esimerkki WSDL-käännöksestä Javan kautta
Suorituksen tarvittava tekninen infrastruktuuri	jätetty avoimeksi (IDL olettaa ORB-tuote, CORBA-nimipalvelu)
Toteutuksessa tarvittavat välineet	IDL-kääntäjät, ohjelmointikielet suoritusinfrastruktuurin lisäksi
Suhde järjestelmän elinkaaren vaiheisiin	
Vaatimukset	Liitteessä A kuvattu vaatimuksia, suunnitteluperiaatteet luvussa 3.2.
Kohdealueen analyysi	Käsitemalli määrittelyt eri osioille erikseen,
Tietosisällön ja toimintojen määrittely	Mahdollista toteuttaa erikseen Messaging API tai Vocabulary API-osat, tai Runtime- ja Browser-osat, tai erilaisia yhdistelmiä näistä.
Tekninen määrittely ja suunnittelu	Sisältää valmiit liittymien määrittelyt IDL:llä
Toteutus	IDL-kieliset liittymät voidaan toteuttaa eri välineillä ja ohjelmointikielillä
Asennus ja käyttöönotto	IDL:lle tavoiteltu erilaisia soveltamistapoja

Ylläpito ja uudet versiot	Standardin ensimmäinen ollut lausunnolla HL7:ssa vuonna 2004.
Arvio levinneisyydestä	
Kansainvälisesti	Uusi standardi, ei vielä laajasti käytössä, tarkoitus tukea HL7 v3 –sovellusten toteuttamista. Mayo Clinic toteuttanut referenssitoteutuksen.
Suomessa	Ei käytössä, ks. yllä
Suhde muihin standardeihin	
Tukee HL7 v3 sanastojen käyttöä sovelluksissa. Käytetyt käsittemallit perustuvat HL7 RIM-mallin käyttöön. Toiminnallisuuden mallina käytetty OMG:n TQS-standardia ja tavoitteena olla sen alijoukko. Lueteltu myös muita asiaan liittyviä määrittäjiä.	
Muuta olennaista	
Standardi on määritelty pyrkien yksinkertaisuuteen rajapintojen hyödyntämisessä. Monissa osissa hyödyntäjän on tunnettava HL7:n sanastokäsitteitä ja –malleja. Määrittämisessä on järkevästi erotettu HL7-viestien käsittelyn operaatiot yleisistä koodistojen hyödyntämisen operaatioista, ja varsinkin jälkimmäiselle asetetut vaatimukset vastaavat pitkälti PlugIT-hankkeen koodistorajapintojen vaatimuksia. Käsittemalli- ja (tekniikkariippumattomalla) rajapintatasolla määrittäminen on kattava, mutta sen pohjalta yhteentoimivien teknisten rajapintojen toteuttaminen vaatii runsaasti tarkennuksia, varsinkin koska IDL:n soveltamisessa ei suositeta suoria käännöksiä CORBA-rajapinnoiksi. Standardin tavoitteena ollut toteutus XML:ään pohjautuen käy ilmi pelkästään lopussa olevasta WSDL-SOAP-esimerkistä.	

PlugIT-hankkeessa oli määritelty ja hyväksytty tekniikkariippumattomat rajapinnat (versio 1) (ks. luku 4), kun HL7 CTS-määrittäminen tuli saataville ja kommentoitavaksi HL7 International:in kautta. CTS:n Vocabulary-osion operaatioille löydettiin vastineet (semanttisesti samaa tavoitetta hoitavat operaatiot) PlugIT-hankkeen Tekniikkariippumattomasta liittymämäärittämisestä, yhtä lukuun ottamatta (areCodesRelated). Teknisiä rajapintoja varten PlugIT-määrittäystä harmonisoitiin CTS:n mukaiseksi, kuitenkin säilyttäen mm. Stakesin koodistopalvelussa käytettyjen sisältöjen vaatimat piirteet. Tehdyt muutokset tekniikkariippumattoman ja teknisen määrittäksen välillä on lueteltu luvussa 4.7.

2.5 Stakesin koodistopalvelin

Osana kansallisen Terveystietojärjestelmän osahanketta 4.1.3 (valtakunnallisen sähköisen sairauskertomuksen käyttöönotto) Stakes on luonut kansallisen koodistopalvelimen. Hankkeen tarkoituksena on ollut aikaansaada sosiaali- terveydenhuollon valtakunnallinen ylläpito- ja jakelupalvelu digitaalisessa muodossa olevia terminologisia järjestelmiä varten (mm. termit, indikaattorit, sanastot, luokitukset ja näihin liittyvät koodistot) (Lehtonen 2003). Hankkeeseen liittyen on määritelty koodistojen siirron ja kyselyn XML-sanomat (Koodistopalvelu 2003). Koodistopalvelu on myös web-käyttöliittymän kautta käytettävissä osoitteessa <http://www.stakes.fi/koodisto/>.

Lähteessä (Ensio, Ruotsalainen 2004, liite 3) on lueteltu suuri määrä terveydenhuollon koodistoja, joita Suomessa hyödynnetään. Stakesin koodistopalvelimelle on suunniteltu seuraavia koodistoja ja luokituksia (Lehtonen 2003):

Ensi vaiheessa

- ICD-10 (Stakes)
- Pohjoismainen leikkaustoimenpideluokitus (Stakes)
- Suun terveydenhuollon toimenpideluokitus (Stakes)
- Radiologinen tutkimus- ja toimenpideluokitus (Suomen kuntaliitto/Qualisan Oy)
- Laboratoriotutkimusnimikkeistö (Suomen kuntaliitto/Qualisan Oy)

- Fysioterapianimikkeistö (Suomen kuntaliitto/Qualisan Oy)
- Toimintaterapianimikkeistö (Suomen kuntaliitto/Qualisan Oy)
- Kustannus Oy Duodecimin synonyymisanasto
- HL7 2.3 mukaisia luokituksia (HL7 Finland ry.)
- HILMO-ohjeistus (Stakes)
- Toimipaikkarekisteri TOPI (Stakes)
- Kuntakoodisto (Stakes)
- muut Stakesin valitsevat tilastoluokitukset

Myöhemmin:

- OID-pohjainen sosiaali- ja terveydenhuollon toimipaikkakoodisto
- Puheterapianimikkeistö 1999
- Avohoidon luokitus ICPC
- ICF (Stakes 2003)
- Terveystieteiden sosiaalityön luokitus 1998
- NordDRG 2003
- ATC lääkeanimikkeistö
- MedDRA
- Lomake nimikkeistö 2002
- Pääotsakenimikkeistö 2002
- Otsakenimikkeistö 2002
- HL7 2.3 koodistoja (10 kpl)
- YSA
- ICD 10 (engl.)
- ICPC (engl.)
- MeSH (engl.)
- HL7 (engl.)
- NCSP 2003 (engl.)

PlugIT-koodistorajapintojen tietosisältömäärittelyissä on hyödynnetty Koodistopalvelimen XML-siirtosanomille määriteltyjä sisältöjä ja merkkauksia (ks. liite 1, Koodistopalvelu 2003).

2.6 OID-yksilöinti

Koodistojen ja organisaatioiden yksilöinnissä (myös Koodistopalvelussa, Terveystieteiden sähköisessä potilaskertomuksessa ja PlugIT-määrittelyissä) käytetään ISO:n määrittelemää OID-tunnistetta. OID (Object Identifier) on numeerinen merkkijono, jolla voidaan yksikäsitteisesti yksilöidä asioita. Kukin organisaatio, joka on saanut OID-numeron, voi laajentaa OID-hierarkiaa oman numeronsa alla haluamallaan tavalla. OID-tunnisteita käytetään mm. eri tyyppisissä hakemistoissa, sertifikaateissa ja dokumenttien yksilöinnissä.

OID-tunnukset sijaitsevat hierarkisessa järjestelmässä, jonka alimmissa solmuissa ovat ISO (<http://www.iso.ch/>) ja ITU (<http://www.itu.ch/>). Nämä organisaatiot delegoivat OID-puun alihaarojen hallintaa muille organisaatioille, jotka delegoivat niitä eteenpäin. OID-tunnisteita käytetään eri protokollissa, tietorakenteissa ja sovelluksissa tarkemmin sovittavilla tavoilla.

OID-tunnisteiden on tarkoitus olla globaalisti yksilöiviä. Niitä muodostetaan lisäämällä yksikäsitteiseen organisaatiokohtaiseen numeeriseen merkkijonoon (esim. 1.3.5.7.9.24.68) lisää numeroita yksikäsitteisellä tavalla (esim. 1.3.5.7.9.24.68.1, 1.3.5.7.9.24.68.2, 1.3.5.7.9.24.68.1.1, jne.). OID-tunnisteen pituudelle ei ole määritelty ylärajaa. OID-tunnusta on tarkoitus käyttää pelkästään vertaamaan, "onko kyseessä sama asia": merkitystä hierarkiassa navigoinnille ja riippuvuuksille

ylemmän tason tunnuksesta tai sovittuja nimiä tietyllä OID-tasolla esitettävillä asioille ei ole määritetty OID-mekanismin osana.

Sosiaali- ja terveydenhuollon alueellisissa tietojärjestelmäratkaisussa suositellaan OID-koodiston käyttöä yksilöinnissä. OID- järjestelmän avulla yksilöidään vastuutaho sekä vastuutahon hallitsevat kohteet. Vastuutahot eli rekisterinpitäjät laativat, ylläpitävät ja julkaisevat muille osapuolille tietojen vaihdossa käyttämänsä koodiluettelot. Luettelot tullaan julkaisemaan Stakesin koodistopalvelun kautta. Koodistopalvelussa eri koodistot yksilöidään OID-tunnisteiden avulla. Esimerkkejä Suomessa käytettävästä koodistojen OID-yksilöinnistä on esitetty mm. raportissa (Ruotsalainen, Ensio 2003).

Koodistojen yksilöinti

Alkuperäiset HL7 koodistot

2.16.840.1.113883.5.n	HL7 version 2.x table
2.16.840.1.113883.5.1001	Service mood code
2.16.840.1.113883.5.1002	Service relationship type code
2.16.840.1.113883.5.1003	Service actor type code
2.16.840.1.113883.6.n	External coding systems
2.16.840.1.113883.6.1	LOINC
2.16.840.1.113883.6.2	ICD 9 CM
2.16.840.1.113883.6.5	SNOMED
2.16.840.1.113883.6.8	Unified Code for Units of Measure

Paikallistetut HL7 koodistot

1.2.246. 777.5.n	Fin-HL7 version 2.x table
1.2.246. 777.5.1001	Fin-Service mood code
1.2.246. 777.5.1002	Fin-Service relationship type code
1.2.246. 777.5.1003	Fin-Service actor type code

Kansalliset sosiaali- ja terveydenhuollon luokitukset ja nimikkeistöt:

1.2.246.537.6.1.1996	ICD 10 1996
1.2.246.537.6.2.1996	Pohjoismainen leikkaustoimenpiteiden nimikkeistö 1996
1.2.246.537.6.3.2002	Laboratorion tutkimusnimikkeistö 2002
1.2.246.537.6.4.2003	Radiologinen tutkimus- ja toimenpidenimikkeistö 2003
1.2.246.537.6.5.2000	Fysioterapianimikkeistö 2000
1.2.246.537.6.6.1999	Puheterapianimikkeistö 1999
1.2.246.537.6.7.1998	Terveydenhuollon sosiaalityön luokitus 1998
1.2.246.537.6.8.2002	Toimintaterapianimikkeistö 2002
1.2.246.537.6.9.2000	NordDRG (Diagnosis Related Groups) 2000
1.2.246.537.6.10.2002	ATC-lääkenimikkeistö
1.2.246.537.6.11.2002	Avohoidon luokitus ICPC
1.2.246.537.6.12.2002	Lomake nimikkeistö 2002
1.2.246.537.6.13.2002	Pääotsake nimikkeistö 2002
1.2.246.537.6.14.2002	Otsake nimikkeistö 2002

Organisaation yksilöinti

Valtakunnallinen yritys ja yhteisötunnus (ei ole toistaiseksi toteutettu)

1.2.246.%y_tunnus.Y_tunnus

3 INTEGROINTIVAATIMUKSET

Tässä luvussa esitellään PlugIT:in koodistorajapintojen määrittelytyön tavoitteet ja tarkennetut vaatimukset. Luku on sisällöllisesti PlugIT-projektissa hyväksytyn ja julkistetun dokumentin ”Koodistorajapinnat: Integroitivaatimukset ja Tekniikkariippumattomat liittymämäärittelyt, versio 1, 11.11.2003” mukainen, mutta version 1 dokumentin taustasioita on siirretty tämän raportin lukuun 2. Määrittelyksen tekniseen versioon 2 (luku 5) tehdyt tarkennukset ja muutokset on koottu lukuun 4.7.

3.1 Tavoitteet ja päärajaukset

Tavoitteet

1. Liittymämäärittely, jolla koodiston tarjoava sovellus (esim. potilashallinnon ydinjärjestelmä tai alueellinen koodistopalvelin) tarjoaa keskeiset koodistopalvelut niitä tarvitseville sovelluksille (luvut 4 ja 5).
2. Sovellusten tarvitsemien koodistopalveluiden ohjelmallisten rajapintojen määrittely yhdenmukaisella tavalla (luvut 4 ja 5).
3. Yhteensopivuus Stakesin koodistopalvelimen sisältöratkaisujen kanssa, kansallisten koodistojen käyttöönotettavuuden edistäminen (luvut 2.5 ja 2.6).
4. Ratkaisun yleisyys: eri koodistoja (mukaan lukien organisaatio) voidaan hyödyntää saman ratkaisun avulla, ja määrittelyjen mukainen ratkaisu on toteutettavissa eri sovelluksiin ja eri ympäristöihin.
5. Määrittelyjen tarkentaminen teknisillä ratkaisuilla (luku 5)
6. Määrittelyjen pilotointi koodistopalveluja tarjoavan, niitä käyttävän ja ratkaisun tilaavan osapuolen kanssa tai referenssitoteutuksen tuottaminen projektissa. (luku 6).
7. Mahdollisten jatkokehityskohteiden tunnistaminen, mm.määrittelyjen kattavuuden lisääminen kohti edistyneempiä terminologiajärjestelmäpalveluita (luku 5.11).

Päärajaukset

1. Tämä dokumentti ja PlugIT-projektin koodistomäärittelyt kohdistuvat pääosin tietojärjestelmiin, jotka toimivat tietyn terveydenhuollon organisaation sisällä (PlugIT-projektin rajausten mukaisesti). Näin ollen se vastaa osin esim. koodistopalvelin-projektin ”sisäinen käyttö (atk-järjestelmät)” –käyttötapaa. Alueellinen tai valtakunnallinen käyttö (esim. HILMO, lähetehoitopalautte, viitetietokannat) eivät ole suoraan määrittelyjen kohteena, mutta määrittelyjä voidaan hyödyntää myös näiden käyttötapojen yhteydessä, ja erityisesti aluetietojärjestelmien vaatimuksia otetaan huomioon työasema- ja web-sovellusten ja ydinjärjestelmien lisäksi.
2. Koodistomäärittelyissä ei tuoteta tietokantaintegraatiota eikä dokumenttipohjaista koodistojen siirtoa tai kopiointia sovelluksesta toiseen, joita varten on Suomessa valmiita tai määrittelyvaiheessa olevia ratkaisuja. Koodistomäärittelyt perustuvat koodistopalveluiden määrittelemiseen siten, että samoja koodistoja ei tarvitse kopioida osallistuviin järjestelmiin ja että palvelut ovat samalla rajapinnalla käytettävissä eri järjestelmissä.
3. Määriteltävät palvelut ovat ohjelmistorajapintojen tarjoamia, ja eivät esim. suoraan näytä ohjelmiston käyttäjälle käyttöliittymää (esim. valintalistaa). Palautettujen tietojen hyödyntäminen (näyttäminen käyttäjälle, muu jatkokäsittely) on palvelua käyttävän sovelluksen vastuulla.

4. Määrittelyt sisältävät vain koodistopalveluiden hyödyntämiseen tarvittavia operaatioita. Koodistojen tai koodiarvojen sisällön päivitys ei kuulu tämän määrittelyn piiriin. Paikallisten koodien ylläpito ja lisäys voi olla mahdollista palvelun tarjoavan tuotteen kautta, ja se voi myös tarjota rajapintoja ylläpitoa varten, mutta nämä rajapinnat eivät ole toistaiseksi tämän määrittelyn kohteena.

Luvun 3 loppuosassa kuvataan PlugIT-projektissa määriteltävien koodistorajapintojen määrittelyyn kohdistuvat vaatimukset. Osio sisältää tarkennetun tavoitemäärittelyn tuotettaville koodistorajapinnoille. Siinä tunnistetaan joukko vaatimuksia ja perusrajoituksia, joiden tarkempia ratkaisuja kuvataan seuraavissa määrittelyissä, joita ovat Tekniikkariippumattomat (luku 4) ja Tekniset liittymämäärittelyt (luku 5). Pilotointiin liittyvät ja paikalliset erityisvaatimukset ja –ratkaisut eivät sisälly tähän dokumenttiin, joka kuvaa yleisiä, moniin sovelluksiin ja käyttötilanteisiin sovellettavissa olevia vaatimuksia ja ratkaisuja.

Vaatimusten lähteinä on käytetty etenkin eri osapuolten kanssa käytyjä keskusteluita, TQS-standardia, Ydinpalveluiden integrointivaatimuksia sisältävää dokumenttia, koodistopalvelimen tietomäärittelyjen mahdollistamia toimintoja (ks. osio 2.4) (TQS 2000, Sormunen ym. 2004a, Koodistopalvelu 2003) ja eri määrittelyversioihin sähköpostilla ja puolivuotisseminaarissa saatuja kommentteja. CTS-standardin aiheuttamia muutoksia määrittelyyn on kuvattu luvussa 4.7.

3.2 Osallistuvat ohjelmistot ja sovellusaluevaatimukset

Vaatimukset kohdistuvat pääosin koodistopalveluita tarjoavaan sovellusosaan (*koodistotarjoaja*). Vastuu eri toimintokokonaisuuksista kohdistuu joko siihen tai koodistopalveluita käyttävään sovellusosaan (*koodistohyödyntäjä*). Mahdollisia koodistotarjoajia ovat mm. sairaaloissa potilashallinnon ydinjärjestelmät kuten Oberon, Effica ja Pegasos (tai erityinen koodistopalvelu), tai alueellisella tasolla aluetietojärjestelmät tai alueelliset koodistopalvelimet. Koodistohyödyntäjiä ovat lukuisat sovellukset (esim. erikoisala- tai tautikohtaiset sovellukset) eri terveydenhuollon yksiköissä.

Seuraavan vaatimusluettelon avulla voidaan eri vaatimuksia listata ja priorisoida otsikkotasolla. Vaatimusten tarkemmat kuvaukset ovat osiossa 3.5 (viite vaatimuksen perässä viittaa vaatimustunnistukseen osiossa 3.5).

Yleiset ja sovellusalueen vaatimukset ja suunnittelukäytännöt

- Koodistoon ja yksittäiseen koodiarvoon liittyvän tiedon ja toiminnallisuuden erottaminen toisistaan K1.1
- Koodistopalvelin-tietosisällön hyödyntäminen K1.2
- Koodiston palveluiden (operaatioiden) ja tietosisällön erottaminen joustavasti K1.3
- Koodistojen keskitetty ylläpito palvelussa, ei kokonaisten koodistojen kopiointia hyödyntäviin järjestelmiin K1.4
- Mahdollisimman kevyt minimitaso olemassa olevien sovellusten liittämisen helpottamiseksi K1.5

3.3 Koodistopalvelut

Minimitasolla riittävät seuraavat piirteet K1.5:

- koodistohyödyntäjän on tiedettävä, mitä koodistotarjoajan koodistoa se hyödyntää (tunnettava koodiston tunniste)
- koodiston saanti käyttöön koodiston tunnisteiden perusteella
- koodin saanti käyttöön koodistosta koodiarvolla (koodin tunnisteella)
- koodistohyödyntäjä voi listata tuntemansa koodiston koodit (koodiarvo ja selite) koodistohyödyntäjän palveluiden avulla (tällöin on myös mahdollista esim. näyttää arvoista valintalista käyttäjälle tai tallettaa koodiston sisältö hyödyntäjän tietokantaan, tms.) K2.3
- koodistohyödyntäjä saa tiettyä koodiarvoa vastaavan (oletus)selitteen koodistotarjoajalta K4.1
- koodien hakeminen tarkan selitteen avulla K2.5

Muita vaatimuksia:

Koko palvelun toiminnallisuus:

- Palvelun sisältämien koodistojen listaus (metatiedot) K3.1
- Palvelun hallinnollisten tietojen hakeminen (tuote, toteuttaja, versio, yhteystiedot) K3.2

Koodistoihin ja koodiarvoihin liittyvä toiminnallisuus:

- Koodistojen yksilöinti OID-tunnusten avulla K2.2
- Koodiston hallinnollisten tietojen kysely K2.4
- Koodien hakeminen osittain annetun käsitteen avulla K2.6
- Valintalistan luominen K2.7
- Kentän validointi, sovelluksen kentässä olevan koodin oikeellisuuden varmistaminen K2.9
- Koodiston ja koko palvelun mahdolliset lisätoiminnallisuudet-kysely (määritellyt lisätoiminnallisuuden tasot) K2.8
- Koodin tunnisteella tapahtuva näytettävän arvon haku K4.1

Koodistoihin laajempi tietosisältö:

- Koodiston sisältämien koodien tietokenttien kysely (metatiedot) K2.1
- Valitun koodin kaikkien kenttien sisällön kysely K4.2
- Valitun koodin nimetyn kentän kysely K4.3
- Koodiston tukemat kielet K2.12
- Koodin tunnisteella tapahtuva näytettävän arvon haku halutulla kielellä K4.4
- Lisätasot ensimmäiseen versioon
- Koodistohierarkian selaamiseen ja rakenteen selvittämiseen tarvittavat operaatiot (esim. organisaatiohierarkia, ICD-luokitus) K2.10
- Koodien hakeminen avainsanojen avulla K2.11
- Paikalliset koodit ja koodien tilatieto K2.15

Mahdolliset myöhemmät laajennukset

- Tarkimman mahdollisen käsitettä vastaavan koodin hakeminen vaiheittain K2.13
- Terminologiapalveluiden käyttö yhdistelmäkäsitteiden (composite concept) validointiin, muuttamiseen jne. K2.14
- Koodin vastaavuudet muissa tuetuissa koodistoissa ja saman koodiston sisällä (mediation) K4.5, CE- ja CD-tyyppisten koodien tuki (MDF 1999).
- Koodin määrittelemän käsitteen esittäminen sovelluksessa annetuilla ehdoilla K4.6

3.4 Tietosisältö

Tietosisällön osalta vaatimuksiin valitaan liitteen 1 listasta (Koodistopalvelu 2003, MDF 1999) sopiva minimitaso: palvelulta jokaiselle koodistolle tai koodiarvolle aina vaadittavat tiedot, jotka otetaan mukaan määrittelyn minimitasoon. Lisäksi liitteen 1 listasta tunnistetaan erikseen monikielisten koodien ja koodistohierarkian hyödyntämiseen ja muihin lisätasoihin tarvittavat tiedot. Nämä vaatimukset kuvataan tässä luvussa tiivistetysti ja luvun 3.5 vaatimusluettelossa.

Tämän lisäksi määritellään vapaasti valittavien tietojen palauttaminen koodistojen ja koodien tietoja palauttavissa operaatioissa. Tätä hyödynnetään, jos palvelu toteuttaa tason, jolla tietoja voidaan vapaasti valita (helpottuu esim. jos palautettavat tiedot ovat XML-muotoisia). Tällöin tarjotaan operaatiot, joilla esim. tässä nimettyjä sisällön osia voidaan pyytää nimellä, tai voidaan pyytää koko koodin tietosisältö. Tietosisällön käytössä pyritään hyödyntämään Stakesin koodistopalvelimen määrittelemiä sisältöjä ja soveltuvin osin myös teknisiä käytäntöjä jatkotyöstyössä.

Tässä mainitut ”eri tasoiset” sisällöt näkyvät tekniikkariippumattomissa määrittelyissä (luku 4) erilaisina tasoina, joista koodistotarjoaja voi toteuttaa haluamansa.

3.4.1 Minimitiedot

Minimitietosisällöksi liitteen 1 listasta muodostuu (K5.1):

Koodistot:

- tunniste: id (suositus OID), jonka avulla voidaan saada lisäksi ainakin
 - koodiston nimi: shortname
 - versio

Koodiarvot:

- tunniste: id (koodiarvo, mikäli yksikäsitteinen koodiston sisällä)
- selite: shortname

Minimitason tiedoilla voidaan toteuttaa CS ja CV-tyyppiset tietokoodaukset (MDF 1999).

Koodiston OID-tunnisteessa voidaan ilmoittaa myös versio (esim. 1.2.246.537.6.17.2002.1).

3.4.2 Monikieliset koodistot, hierarkian käsittely ja edistyneet haut

Eri kielisten selitteiden hyödyntämistä varten minimitietosisällön lisäksi hyödynnetään (K5.2):

Koodistot sekä koodiarvot:

- kieli (language) –tietoa yhdessä määriteltyjen minimitietojen kanssa sekä eri kielisiä selitteitä tai termejä koodiston sisällä

Luokitushierarkioiden käsittelyä varten voidaan hyödyntää (K5.3):

Koodiarvot:

- tunniste: id,
- yläluokka: parentID,
- hierarkiataso: hierarchyLevel,
- leafnode, child - tietoja

Perushakutoimintoja monipuolisemmassa haussa hyödynnetään (K5.4):

Koodiarvot

- avainsanoja (keyword)
- lisäksi koko koodiston sisällöstä voi olla tarpeen hakea (ali)merkkijonoja

On myös tilanteita, joissa pitää saada selville, ovatko koodiarvot paikallisia (alkuperäiseen, esim. kansalliseen koodistoon jälkikäteen lisättyjä) tai merkitty poistetuiksi tai ehdotetuiksi (vai aina käytössä oleviksi). (K2.15). Näissä tilanteissa voidaan hyödyntää status- ja local-tietoelementtejä.

3.4.3 Vapaa tietosisältö

Jotta kaikkia mahdollisia eri koodistojen sisältämiä tietoja voitaisiin hyödyntää, on tarjottava mahdollisuus hakea mitä tahansa koodiston sisältämiä elementtejä. Jos pohjana käytetään esim. Koodistopalvelimen XML-määrittäjiä, pidetään käytäntönä haettavien tietojen nimeämistä ”attribuutin tyyppien nimet” -listan (liite 1) mukaisesti. (K5.5)

3.5 Vaatimusten tarkempi kuvaus

K1.1	Koodistoon ja yksittäiseen koodiin liittyvän tiedon ja toiminnallisuuden erottaminen toisistaan
Kuvaus:	Hyvä suunnittelukäytäntö, mahdollistaa sekä oliopohjaisten että palvelupohjaisten ratkaisujen tuottamisen ja mm. XML:n hyödyntämisen myös eri ”operaatiotekniikoilla”. Määriteltävä kuitenkin riittävä määrä tietoa toiminnan onnistumiseksi.
Kohde:	Tehtävät määritykset
Prioriteetti:	3 toivottava
Tukimateriaali:	
Tyyppi:	laadullinen
Historia:	v0.5
Lähde:	PlugIT

K1.2	Koodistopalvelin-tietosisällön hyödyntäminen
Kuvaus:	Yhdenmukaisuus kansallisen koodistopalvelimen kanssa
Kohde:	Tehtävät määritykset
Prioriteetti:	2 tärkeä
Tukimateriaali:	Koodistopalvelimen määritykset
Tyyppi:	sovellusalue
Historia:	v0.5
Lähde:	PlugIT, koodistopalvelin

K1.3	Palveluiden (operaatioiden) ja tietosisällön erottaminen
Kuvaus:	Edesauttaa ratkaisun joustavuutta ja laajennettavuutta jatkossa
Kohde:	Tehtävät määritykset
Prioriteetti:	3 toivottava
Tyyppi:	laadullinen
Historia:	v0.5
Lähde:	PlugIT

K1.4	Koodistojen keskitetty ylläpito palvelussa, ei kokonaisten koodistojen kopiointia hyödyntäviin järjestelmiin
Kuvaus:	Tavoitteena esim. alueellisesti keskitetty koodistojen ylläpito tai ylläpito vain ydinjärjestelmiin ja koodistojen nopea käyttöönsaanti sovelluksissa. Kopiointi sovelluksiin mahdollista esim. koodistopalvelimen määritysten avulla.
Kohde:	Tehtävät määritykset
Prioriteetti:	2 tärkeä
Tyyppi:	laadullinen
Historia:	v0.7
Lähde:	HUS, Koodistopalvelin (seminaari 28.10.03), PlugIT

K1.5	Kevyt minimitaso
Kuvaus:	Mahdollisimman kevyt minimitaso olemassa olevien vain joitain koodistoja hyödyntävien sovellusten liittäminen helpottamiseksi
Kohde:	Tehtävät määritykset, operaatioiden ryhmittely
Prioriteetti:	1 kriittinen
Tukimateriaali:	
Tyyppi:	laadullinen, käyttöönottokynnys
Historia:	v0.7
Lähde:	PlugIT, TE, MD (seminaari 27.10.03)

K2.1	Koodiston sisältämien koodien tietokenttien kysely (metatiedot)
Kuvaus:	Koodistohyödyntäjä voi kysellä, mitä tietokenttiä kukin koodisto sisältää. Tarpeen tuetun sisällön selvittämiseksi.
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	2 tärkeä
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	PlugIT

K2.2	Koodistojen yksilöinti OID-tunnusten avulla
Kuvaus:	Koodiston yksilöinnin tavasta sopiminen, noudatetaan muodostuvaa kansallista käytäntöä
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	2 tärkeä
Tukimateriaali:	Ks. liite 1
Tyyppi:	toiminnallinen, tietosisältö, sovellusalueen vaatimus
Historia:	v0.5
Lähde:	Koodistopalvelu, PlugIT

K2.3	Koodiston sisältämien koodiarvojen listaus
Kuvaus:	Tärkeä toiminnallisuus koodistojen hyödyntämiseksi. Arvot listataan palvelun määrittelemässä järjestyksessä. Arvoja voi listata myös pyydetyn määrän kerrallaan.
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	1 kriittinen
Tukimateriaali:	TQS
Tyyppi:	toiminnallinen, operaatiot, laadullinen (maksimimäärä-suorituskyky)
Historia:	v0.5
Lähde:	TQS

K2.4	Koodiston hallinnollisten tietojen kysely
Kuvaus:	Koodistohyödyntäjä saa palvelulta ihmisen ymmärrettävässä muodossa koodiston ylläpitäjän ym. hallinnolliset tiedot.
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	3 toivottava
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5, siirretty perustasolle v0.9
Lähde:	Koodistopalvelu, Mediconsult

K2.5	Koodien hakeminen tarkan käsitteen avulla
Kuvaus:	Koodistohyödyntäjä voi etsiä palvelun koodistosta tarkkaa osumaa selitteeseen (osumia voi olla useita).
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	1 kriittinen
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5, v0.7 muutettu kriittiseksi
Lähde:	TQS

K2.6	Koodien hakeminen osittain annetun käsitteen avulla
Kuvaus:	Koodistohyödyntäjä voi etsiä palvelun koodistosta hakuavaimella, palvelu palauttaa koodit, joista hakuavain löytyy selite-osasta (osumia voi olla useita).
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	1 kriittinen
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	TQS

K2.7	Valintalistan luominen
Kuvaus:	Koodistohyödyntäjä saa kooditarjoajalta valintalistan (halutuilla koodin kentillä), käyttökontekstista riippuva valintalistan luonti
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	1 kriittinen
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5 poistettu v0.6, ks Luku 4.6
Lähde:	TQS

K2.8	Koodiston lisätoiminnallisuudet-kysely
Kuvaus:	Koodistohyödyntäjä voi kysyä, mitä lisätoiminnallisuuksia palvelu tarjoaa tietyille koodistolle. Tarvitaan, jos palveluun halutaan laajennettavaa lisätoiminnallisuutta.
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	3 toivottava
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	PlugIT, Mediconsult

K2.9	Kentän validointi
Kuvaus:	Koodistohyödyntäjä voi varmistua kentässä olevan koodin oikeellisuudesta koodiarvon avulla
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	3 toivottava
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5, tarkennettu v0.6 koodiarvon avulla (TQS)
Lähde:	TQS

K2.10	Hierarkian selaamisen operaatiot
Kuvaus:	Koodistohyödyntäjä voi selata ja selvittää koodiston rakennetta palvelun operaatioiden avulla hierarkkisessa koodistossa (esim. ICD-luokitus), jatkossa tarkennettava mm. onko organisaatio-OID-arvoilla suhdetta organisaatiohierarkiaan
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	2 tärkeä
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	TQS, Mediconsult, Koodistopalvelin, Fujitsu

K2.11	Koodien hakeminen avainsanojen avulla
Kuvaus:	Selitteen lisäksi koodistohyödyntäjä voi käyttää koodeille määriteltyjä avainsanoja (esim. mittayksikkö) koodien hakemiseen
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	3 toivottava
Tukimateriaali:	Voidaan liittää hakemiseen selitteistä (vaatimus K2.5, K2.6)
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	TQS, Koodistopalvelu

K2.12	Koodiston tukemat kielet
Kuvaus:	Koodistohyödyntäjä voi selvittää koodistokohtaisesti, mitä selitteiden kieliä koodisto sisältää. Tarvitaan, jos käytetään monikielisiä koodistoja ja ei nojaututa ”oletuskielen” käyttöön.
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	3 toivottava
Tukimateriaali:	ks. K4.4, K5.2
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	Koodistopalvelu, Mediconsult, Fujitsu

K2.13	Tarkimman mahdollisen käsitettä vastaavan koodin hakeminen vaiheittain
Kuvaus:	Selvitettävä , mitkä vaiheet ovat (käytetäänkö esim. koodiston sisäisiä ristiviittauksia), vai riittääkö K2.10 mukainen hierarkkinen tarkennus).
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	3 toivottava
Tukimateriaali:	Liittyy (onko korvattavissa) vaatimukseen K2.10
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5 siirretty jatkokehitysideoihin v0.6, ks. luku 4.6
Lähde:	TQS

K2.14	Terminologiapalveluiden käyttö yhdistelmäkäsitteiden validointiin, muuttamiseen jne.
Kuvaus:	Selvitettävä, halutaanko ja miten yhdistelmäkäsitteitä (composite concept) hyödyntää.
Kohde:	Koodistorajapinta, koodisto
Prioriteetti:	3 toivottava
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5 siirretty jatkokehitysideoihin v0.6
Lähde:	TQS

K2.15	Paikalliset koodit ja koodien tilatieto
Kuvaus:	Osa koodeista voi olla paikallisesti lisättyjä, osa voi olla merkitty poistetuiksi (esim. paikalliset korvautuneet valtakunnallisilla) tai ehdotetuiksi.
Kohde:	Koodistotarjoaja ja –rajapinta, tietosisältö ja toiminnallisuus
Prioriteetti:	2 tärkeä
Tukimateriaali:	
Tyyppi:	toiminnallinen, tietosisältö
Historia:	v0.7, uusi taso: status
Lähde:	PSSHP (seminaari 28.10.03), Mediconsult

K3.1	Palvelun sisältämien koodistojen listaus (metatiedot)
Kuvaus:	Koodistotarjoajalta voi kysyä, mitkä koodistot se sisältää, palautetaan koodistojen tunnisteet (OID), versiot ja nimet
Kohde:	Koodistorajapinta, koko palvelu
Prioriteetti:	2 tärkeä
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	PlugIT, MDF

K3.2	Palvelun hallinnollisten tietojen hakeminen
Kuvaus:	Koodistopalvelulta voi kysyä palvelun ja sen toteuttajan tietoja (tuote, toteuttaja, versio, yhteystiedot)
Kohde:	Koodistorajapinta, koko palvelu
Prioriteetti:	2 tärkeä (v0.9)
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5, siirretty perustasolle v0.9
Lähde:	TQS, Mediconsult, Fujitsu

K4.1	Koodin tunnisteella tapahtuva näytettävän arvon haku
Kuvaus:	Koodistotarjoaja palauttaa koodiarvon perusteella selitteen
Kohde:	Koodistorajapinta, koodi
Prioriteetti:	1 kriittinen
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	TQS, koodistopalvelu, PlugIT

K4.2	Valitun koodin kaikkien kenttien sisällön kysely
Kuvaus:	Koodistotarjoaja palauttaa koodiarvon perusteella koodin kaikki tiedot. Tarvitaan joka tapauksessa, jos käsitellään muutakin sisältöä kuin koodin ja selitteen sisältäviä koodistoja.
Kohde:	Koodistorajapinta, koodi
Prioriteetti:	1 kriittinen
Tukimateriaali:	
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	TQS

K4.3	Valitun koodin nimetyn kentän kysely
Kuvaus:	Koodistotarjoaja palauttaa koodiarvon perusteella koodin nimetyn kentän tiedot. Voidaan tarjota kaikkien kenttien sisällön kyselyn lisäksi.
Kohde:	Koodistorajapinta, koodi
Prioriteetti:	2 tärkeä
Tukimateriaali:	ks. K4.2
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	TQS

K4.4	Koodin tunnisteella tapahtuva näytettävän arvon haku halutulla kielellä
Kuvaus:	Tarvitaan, jos tuetaan useita selitteiden kieliä, ks K2.12
Kohde:	Koodistorajapinta, koodi
Prioriteetti:	2 tärkeä
Tukimateriaali:	ks. K2.12, K5.2
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5
Lähde:	Koodistopalvelu

K4.5	Koodin vastaavuudet muissa tuetuissa koodistoissa ja saman koodiston sisällä (mediation)
Kuvaus:	Edistynyt koodistojen välinen terminologioiden mediaatio, toteutuskohtainen?
Kohde:	Koodistorajapinta, koodi
Prioriteetti:	2 tärkeä (v0.9) 3 toivottava (v0.5)
Tukimateriaali:	ks. K2.12
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5 siirretty jatkoideoihin v0.6 nostettu tärkeäksi tulevissa versioissa ja myös koodiston sisäiset suhteet v0.9
Lähde:	TQS, Koodistopalvelu, Mediconsult

K4.6	Koodin määrittelemän käsitteen esittäminen sovelluksessa annetuilla ehdoilla
Kuvaus:	Edistynyt, mm. taivutus- ja monikkomuodot huomioon ottava koodin esitystavan muokkaaminen koodistotarjoajan palvelulla
Kohde:	Koodistorajapinta, koodi
Prioriteetti:	3 toivottava
Tukimateriaali:	ks. K2.12
Tyyppi:	toiminnallinen, operaatiot
Historia:	v0.5 siirretty jatkoideoihin v0.6
Lähde:	TQS

K5.1	Minimitietosisällön määrittely
Kuvaus:	Tarvittavat tiedot yksinkertaisten koodipalveluiden rakentamiseen sisältävät koodistoista tunnisteiden, nimien ja version, ja koodeista tunnisteiden (koodiarvon) ja selitteen
Kohde:	Koodistorajapinta ja –tarjoaja, tietosisältö
Prioriteetti:	1 kriittinen
Tukimateriaali:	ks. K1.5
Tyyppi:	toiminnallinen, tietosisältö
Historia:	v0.6
Lähde:	PlugIT, MDF, TE, MD

K5.2	Eri kielten tukeen vaadittava tietosisältö
Kuvaus:	Palvelun on sisällettävä kielitieto sekä koodistoille että koodiarvoille, jos eri kielisiä selitteitä ja/tai koodistoja halutaan hyödyntää
Kohde:	Koodistotarjoaja ja –rajapinta, tietosisältö
Prioriteetti:	2 tärkeä
Tukimateriaali:	ks. K2.12, K4.4, ISO 639-standardi (kielten lyhenteet)
Tyyppi:	toiminnallinen, tietosisältö
Historia:	v0.6
Lähde:	Koodistopalvelu, PlugIT

K5.3	Hierarkian käsittelyyn vaadittava tietosisältö
Kuvaus:	Palvelun on sisällettävä tietoa koodistojen hierarkkisista suhteista (yläluokka, hierarkiataso, tieto lehtisolmusta, alaluokat), jos palvelu tukee luokitushierarkioiden selausta ja rakenteen selvittämistä
Kohde:	Koodistotarjoaja ja –rajapinta, tietosisältö
Prioriteetti:	2 tärkeä
Tukimateriaali:	
Tyyppi:	toiminnallinen, tietosisältö
Historia:	v0.6
Lähde:	Koodistopalvelu, PlugIT

K5.4	Monipuolisten hakujen tietosisältö
Kuvaus:	Perushakujen lisäksi, avainsanojen käyttämiseksi hauissa palvelussa ja koodistoissa on oltava määritellyt avainsanat. Myös koodiston kaikesta sisällöstä voi olla tarpeen tehdä merkkijonohakuja.
Kohde:	Koodistotarjoaja ja –rajapinta, tietosisältö
Prioriteetti:	2 tärkeä
Tukimateriaali:	
Tyyppi:	toiminnallinen, tietosisältö
Historia:	v0.6
Lähde:	Koodistopalvelu, PlugIT

K5.5	Vapaa tietosisältö
Kuvaus:	Joissakin koodistoissa on suuri määrä tietokenttiä. Minkä tahansa nimetyn kentän arvosta saanti voi olla tarpeen toteutuksissa.
Kohde:	Koodistotarjoaja ja –rajapinta, tietosisältö
Prioriteetti:	2 tärkeä
Tukimateriaali:	
Tyyppi:	toiminnallinen, tietosisältö
Historia:	v0.6
Lähde:	Koodistopalvelu, PlugIT

4 TEKNIKKARIIPPUMATTOMAT LIITTYMÄ-MÄÄRITTELYT

Luvussa 4 kuvataan määriteltävän ratkaisun ja rajapintojen sisältö (tieto ja toiminnallisuus) tekniikkariippumattomalla tasolla.

Luvun 4 pohjalta voidaan tuottaa valituilla integrointitekniikoilla tarkemmat, teknisen integroinnin toteutuksen eri järjestelmiin mahdollistavat määrittelyt. Luvussa 5 on kuvattu (luvun 4.7 tarkennuksin) http- ja XML-rajapinnoiksi tarkennettu rajapinta.

Tekniikkariippumaton liittymämäärittely vastaa rajattuun integrointitarpeeseen ja määrittelee tämän tarpeen ratkaisun sisällön ja peruseriaatteen sitomatta sitä yksittäisiin liittymä- tai toteutus-tekniikoihin.

4.1 Integrointiperiaatteet ja sovellusten vuorovaikutus

Määriteltävässä ratkaisussa osapuolina ovat:

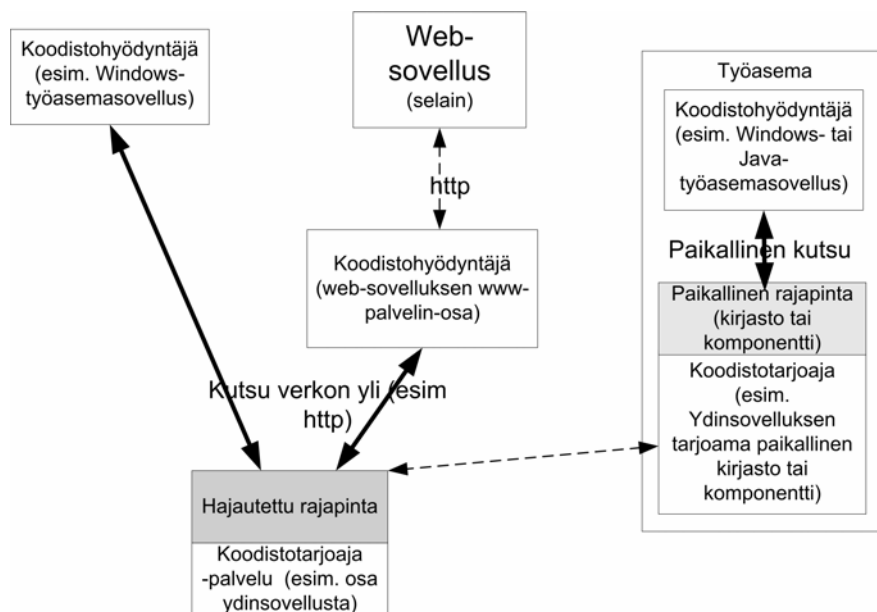
1. *Koodistotarjoaja*, sovellus tai komponentti, joka tarjoaa määritellyt rajapinnat tai rajapintajoukot ja vastaa ratkaisussa sen vastuulle määritellyistä seikoista.
2. *Koodistohyödyntäjä*, sovellus tai komponentti, joka käyttää koodistotarjoajan tarjoamia palveluita.

Ratkaisun arkkitehtuurilliset perusratkaisut ja pääperiaatteet:

1. Määriteltävät ratkaisut kohdistuvat rajausten mukaisesti pääasiassa *ohjelmistorajapintoihin*, joiden tarjoamisesta vastaa koodistotarjoaja. Koodistohyödyntäjä käyttää näitä rajapintoja ja vastaa määritellyissä ratkaisuissa sen vastuulle annetuista seikoista.
2. Kyseessä on *palvelupohjainen* integrointimalli, jossa sovellukset jakavat yhteisiä palveluita.. Tämä saavutetaan määrittelemällä operaatioita, joita voidaan käyttää (siis integroida) useissa eri sovelluksissa, ja sopimalla infrastruktuurista, jota käytetään näiden operaatioiden kutsumiseen. Koodistohyödyntäjä voi nojautua koodistotarjoajan palveluihin, eikä sen tarvitse kopioida kokonaisia koodistoja esim. omaan tietokantaansa (vaatimus K1.4, ks. myös koodistohyödyntäjä-esimerkit, luku 4.3.2).
3. Tällä tasolla määritellyt rajapinnat voidaan tarkentaa toteutettavaksi ja toteuttaa joko (tasot on määritelty viitearkkitehtuurissa dokumentissa Mykkänen ym. 2004c):
 - hajautettuna (kutsutaan verkon yli): hajautetussa käytössä integrointipiste on palvelun toteutuksen kannalta esim. sovellus- tai web-palvelimella, (enterprise-taso, palvelin) ja palveluhyödyntäjän kannalta tyypillisesti joko palvelin- (enterprise) tai työasema (workspace, työpöytä)-tasolla,
 - tai paikallisena (kutsutaan samassa osoiteavaruudessa), sekä palvelutarjoajan että hyödyntäjän esim. samalla työasemalla (workspace, työpöytätasolla).Hajautettu käyttö on ensisijainen tavoite. Käyttötapa määritellään kussakin teknisessä liittymämäärittelyssä (määrittelyn seuraava vaihe), joita voi olla eri tekniikoille (ks. Kuva 4.1).
4. Luku sisältää operaatioiden määrittelyt sitomatta niitä tiettyyn integrointitekniikkaan. Teknisenä infrastruktuurista ja käytetyistä tekniikoista sovitaan teknisissä liittymämäärittelyissä. Jos tekniseen määrittelyyn valitaan avoimet tekniikat (kuten XML ja http tai SOAP), rajapintoja voidaan kutsua eri tekniikoilla sisäisesti toteutetuista sovelluksista.

5. Ratkaisun periaatteena on koodistohyödyntäjän kannalta synkroninen pyyntö/vastaus, jossa koodistohyödyntäjä saa lähettämäänsä kyselyyn välittömästi vastauksen (tai virheen) koodistotarjoajalta.

Koodistorajapinnat voidaan toteuttaa tilattomina palveluina, mikä tarkoittaa sitä että kaikki tarvittava tieto (esim. myös koodiston tunniste) liikkuu parametrina jokaisessa ohjelmistokutsussa koodistotarjoajalle. Toinen vaihtoehto on, että koodistotarjoaja säilyttää tiedon esim. valitusta koodistosta, tarjoten koodistoja (ja koodiarvoja) olio-tyyppisillä rajapinnoilla. Näiden määrittelyiden tarkentaminen tilattomiksi palveluiksi tapahtuu määrittelemällä eri rajapintatasoille (code, codeset, codeservice) tarvittavat lisäparametrit. Tarkentaminen oliopohjaisiksi rajapinnoiksi onnistuu määrittelemällä palvelun vastuu tilan säilyttämisestä ja poistamalla säilytettävät tunnistetiedot rajapintojen operaatioista. On mahdollista että koodistotarjoaja tarjoaa sekä hajautetun, tilattoman palvelun (yhdellä avoimella tekniikalla) että paikallisia oliopohjaisia kirjastoja (tai välineitä) eri tekniikoille (esim. erikseen eri tekniikoilla kuten Java- ja Windows-kirjastot).



Kuva 4.1. Esimerkkejä määritellyn palvelun (harmaalla) teknisistä toteutuksista ja käyttötavoista.

Tässä luvussa määritellyissä rajapinnoissa:

- operaatioiden määrittelyssä hyödynnetään soveltuvin osin TQS-standardia (TQS 2000)
- tietosisällön määrittelyissä ja tietoelementtien nimissä hyödynnetään Stakesin koodistopalvelimen, TQS-standarin ja HL7 Message Development Framework- määrittelyksen tietosisältöä koodistoihin liittyen (Koodistopalvelu 2003, TQS 2000, MDF 1999)
- tietotyyppien määrittelyssä käytetään HL7-CEN harmonisoitua Health Informatics - Data Types -määrittelyä (CEN 2003),
- määritellyt operaatiot ja tietosisältö ovat peräisin määritellyistä projektin osapuolten vaatimuksista tai muuten tarpeelliseksi havaituista piirteistä (luku 3).

Käytettävä käsitelmä on hyvin yksinkertainen ja joustava:

koodistotarjoajan tarjoama *koodipalvelu* koostuu useista *koodistoista*, jotka puolestaan koostuvat *koodeista*. Koodit koostuvat vähintään *koodiarvoista* ja *selitteistä*, ja lisäksi niillä voi olla muita tietokenttiä. *Lisätasoilla* mm. koodien *hierarkkiset suhteet* ja viittaukset muissa koodistoissa sijaitseviin koodeihin voidaan lisätä malliin. Koodisto/koodi-malli on Stakesin

koodistopalvelimen XML-määritysten mukainen, jossa koodistoilla ja koodeilla voi lisäksi olla runsaasti samoja sisällöllisiä osia.

Valtakunnallisten koodistojen tunnisteenä käytetään *OID-koodia* (vaatimus K2.2, luku 2.6), ja on suositeltavaa käyttää OID-koodeja myös paikallisten koodistojen tunnistamiseen, mutta kooditarjoaja voi ymmärtää myös paikallisia tunnisteita. OID-koodiin voidaan määritellä myös koodiston vuosi ja versio (Koodistopalvelu 2003). Esim. koodistotarjoajan tarjoamia koodistoja kysellessä palvelu palauttaa kaikki versiot (kokonaiset OID-polut), joita se tukee. Asiakas voi viitata koodistoon pelkällä koodisto-OID:llä (ilman versiota), jolloin palvelu palauttaa oletusversion (yleensä uusin versio) mukaisia vastauksia.

4.2 Tekniikkariippumattomat operaatiomäärittelyt

Määrittelyksen osia (ja toimintokokonaisuuksia, joita koodistotarjoaja voi tarjota) ovat:

- *minimitaso* – vain keskitetyn koodistopalvelun välttämättömät operaatiot olemassa olevien sovellusten helpon liitettävyyden kannalta (vaatimus K1.5): koodiston ja koodin käyttöön saanti, koodien listaus, minimitietosisältö (koodi ja selite), selitteen saanti koodilla, koodi(e)n saanti selitteellä
- *perustaso* – minimitason lisäksi osamerkkijonolla haku selitteestä ja koodista, tuettujen palveluiden ja koodistojen selville saanti (vaatimus K5.1), palvelun ja koodistojen (vapaaehtoiset) hallinnolliset tiedot (K3.2, K2.4),
- *monikieliset koodistot* (vaatimus K5.2),
- *vapaa tietosisältö* - perustietojen (koodiarvo, selite) lisäksi minkä tahansa koodien sisältämien kenttien käyttö eri operaatioissa (vaatimus K5.5),
- *edistyneet haut* - haku koodiston koko sisällöstä ml. alimerkkijonot, haku avainsanoilla, hakuehtojen kohdistaminen haluttuihin kenttiin (vaatimus K5.4),
- *koodistohierarkian* käsittely (vaatimus K5.3),
- *paikallisten koodien* ja *tilan* huomiointi (vaatimus K2.15) - oletusarvoisesti palvelu palauttaa myös paikalliset ja poistetuiksi merkityt koodit.

Minimitaso vaaditaan kaikilta toteutuksilta. Perustaso on suositeltava minimitason lisäksi. Muut tasot voidaan toteuttaa niiden lisäksi toisistaan riippumatta, ellei toisin mainita. Tasoja ja operaatioita eri tasoille voidaan jatkossa lisätä tarpeen mukaan esim. eri terminologioiden suhteita tai yhdistelmäkäsitteiden käsittelyä varten.

Eri tasojen operaatioita ja parametreja (ks. myös liite 3) voidaan yhdistellä toteutustekniikoiden sallimilla tavoilla teknisissä määrittelyissä, jolloin voidaan korvata osa parametreista oletusarvoilla, ja näin operaatioiden lukumäärä pienenee, ks. luku 4.4 esimerkki.

Koodistotarjoajan palvelut ryhmitellään seuraaviin rajapintoihin:

- koodistopalvelun toteutukseen liittyvät palvelut: *codeservice*-rajapinta - liittyvät itse palvelun toteutukseen ja hyödyntämiseen.
- koodistoon liittyvät palvelut: *codeset*-rajapinta -kohdistuvat yksilöityyn (yhteen) koodistotarjoajan tarjoamaan koodistoon. Tarvittaessa teknisessä toteutuksessa välitetään koodiston tunnistetta (in-) parametrina eri operaatioille.
- koodiin liittyvät palvelut: *code*-rajapinta - liittyvät tunnistettuun ja yksilöityyn koodiin.

Kunkin operaation yhteydessä on mainittu, mihin rajapintaan se kuuluu.

Rajapintoja voidaan yhdistellä ja tarkentaa teknisissä määrittelyissä tarvittaessa, esim. muokkaamalla tietotyyppejä, operaatioita ja niiden ryhmittelyä tekniikalle sopiviksi. Esimerkiksi jos halutaan käsitellä koodi-rajapintaa, voidaan jollakin tekniikalla tarjota ”koodi-olio”, joka saadaan käyttöön koodistopalvelun ja koodiston kautta, toisella tekniikalla käsitellä koodi-rajapintaa antamalla kutsussa parametrina sekä koodipalvelun, koodiston että koodin tunniste. Eri tekniikoilla voidaan tarjota operaatiot, joilla codeservice palauttaa codeset-rajapintansa ja codeset tukemansa codet (rajapintoina) (operaatio esim. codeset.get_code(code_id)). Tämä riippuu kuitenkin valitusta tekniikasta, ja tässä määritelty tunnistetietojen käsittely riittää yksinkertaisissa toteutuksissa vastaavaan toiminnallisuuteen.

Koodistotarjoajan on tarjottava ainakin codeservice, codeset ja code –rajapintojen tärkeimmät operaatiot. Joitakin määrittelyistä operaatioista voitaisiin toteuttaa sekä koodisto- että koodi-rajapintoihin. Usein näissä tapauksissa operaatio on määritelty koodisto-rajapintaan, ja esim. kaikki koodi-rajapintaan määritellyt operaatiot voidaan teknisessä määrittelyssä siirtää helposti koodisto-rajapintaan.

Sekä in- että out-parametrit ovat pakollisia, ellei niitä ole määrittelyssä erikseen merkitty ”vapaaehtoiseksi”. Vapaaehtoisten parametrien käsittely tarkennetaan teknisissä määrittelyissä kullekin tekniikalle sopivaksi. Parametrit, joille on määritelty oletusarvo, korvaa koodistotarjoaja tällä oletusarvolla, jos koodistohyödyntäjä on jättänyt sen määrittelemättä kutsussa.

Kohdassa poikkeukset teksti ”palvelu palauttaa virheen” tarkoittaa sitä, että toiminta tarkennetaan teknisissä määrittelyissä. Tyypillisesti virhe palauttaa esim. virhekoodin ja mahdollisen virheilmoituksen, tai määritellyn virhettä tarkoittavan paluuarvon. Virhetilanteita käsitellään myös luvussa 4.3.1.

Määrittelyn mukaisuutta käsitellään luvussa 4.5.2. Toteutuksiin voidaan määritellä (ja dokumentoida) laajennuksia, joita ei ole määritelty tässä dokumentissa.

Tämän luvun operaatiomäärittelyä on tarkennettu ja operaatioiden määrää vähennetty parametreja yhdistämällä teknisissä määrittelyissä, jonka operaatioiden yhteenveto löytyy myös liitteestä 2.

4.2.1 Minimitaso

Minimitason (vaatimus K1.5) vaatimukset kuvattiin luvussa 3.

Koodiston ja koodin saanti palvelusta

Teknisissä määrittelyissä on määriteltävä (ks. myös tekniset esimerkit, luku 4.4):

- koodiston tunnisteella tapahtuva koodiston käyttö: esim. codeservice: get_codeset – operaatio voi saada parametrina koodiston tunnisteen ja palauttaa rajapinnan käytettävään koodistoon, tai koodiston tunniste voidaan välittää parametrina kussakin kutsussa
- yhden koodin käyttö koodiarvolla: esim. codeset: get_code – operaatio voi saada parametrina koodiarvon ja palauttaa rajapinnan yksittäiseen koodiin, tai koodin tunniste voidaan välittää koodisto-rajapinnassa parametrina

Taulukko 4.1. Minimitason operaatiot.

Koodien haut

Operaatio	codeset: get_concepts_by_text
Käyttötarkoitus	Koodien hakeminen tarkan selitteen avulla (vaatimus K2.5)
Parametrit (in)	key: ST – hakuarvo
Paluuarvot (out)	codes: SET (code_id: ST, value: ST) - Tietojoukko, joka koostuu elementeistä, joissa on koodiarvo sekä selite.
Poikkeukset	Paluuarvo on tyhjä joukko, jos selitettä ei vastaa tarkasti mikään selite koodistossa
Kuvaus	Palvelu palauttaa ne koodit, joiden selite vastaa täsmälleen annettua hakuarvoa, pienet ja isot kirjaimet vastaavat toisiaan
Esimerkki	haku ”lavantauti” palauttaa ICD-10-koodistosta joukon, jossa on ”A01.0””Lavantauti”, mutta ei ”A01.0 +J17.0””Lavantautiin liittyvä keuhkokuume”
Pakollisuus	Pakollinen kaikissa toteutuksissa
Malli	TQS: CodingSchemeVersion: get_concepts_by_text
Lisätietoja	Koska selitteet eivät välttämättä ole yksikäsitteisiä, on palautettava joukko. Jos koodistossa on vain yksikäsitteisiä selitteitä, voidaan teknisissä määrittelyissä määritellä näissä tilanteissa vain yhden koodin palautus mikäli käytettävä tekniikka tukee tällaista joustavuutta.

Koodien listaus

Operaatio	codeset: list_concepts
Käyttötarkoitus	Koodien listaus (vaatimus K2.3, K2.7)
Parametrit (in)	how_many: Int – kerralla palautettavien maksimimäärä (vapaaehtoinen), from: ST – ensimmäisen haettavan koodin koodiarvo (tunniste), tyhjä jos haetaan alusta lähtien
Paluuarvot (out)	more: ST – ensimmäisen vielä jäljellä olevan (palauttamattoman) koodin koodiarvo (tunniste). Jos kaikki palautettu, tyhjä merkkijono. codes LIST (code_id: ST, value: ST) - Tietojoukko, joka koostuu elementeistä, joissa on koodiarvo sekä selite.
Kuvaus	Palvelu palauttaa listan koodiston koodeista koodiarvon määrittelemässä järjestyksessä. Peräkkäisillä hauilla voidaan jatkaa listausta ensimmäisestä vielä palauttamattomasta koodista. Jos how_many-parametri puuttuu, koodistotarjoaja palauttaa toteutuskohtaisen oletusmäärän vastauksia.
Poikkeukset	Jos how_many-parametri ylittää koodistotarjoajan toteutuskohtaisen maksimimäärän, palvelu palauttaa virheen.
Esimerkki	
Pakollisuus	Pakollinen kaikissa toteutuksissa
Malli	TQS: CodingSchemeVersion: list_concepts
Lisätietoja	TQS:ssä käytettävä iteraattori-olio korvattu vastaavilla parametreilla ja paluuarvoilla. Hierarkkisissa koodistoissa suositellaan erityisesti niihin tarkoitettujen operaatioiden käyttöä. Jos halutaan listata vain voimassaolevat koodit, on toteutettava ”status” tason mukaiset palvelut. Lajitteluparametrit (minkä kentän mukaan lajiteltuna tulokset palautetaan) on mukana freeElements-tason list_concepts_and_values-operaatioissa.

Yhden koodin arvon hakeminen (huom. code-rajapinta)

Operaatio	code: get_preferred_text
Käyttötarkoitus	Selitteen hakeminen sovellukseen tietyllä koodilla (vaatimus K4.1)
Parametrit (in)	(ei mitään koodi-rajapinnassa; jos ei koodi-rajapintaa vaan toteutettu koodisto-rajapintaan, koodin tunniste)
Paluuarvot (out)	value:ST - Selite (näytettävä arvo), palvelussa määritellyllä oletuskielellä, jos useita kieliä käytössä
Poikkeukset	Jos annetulla koodiarvolla ei löydy koodia, paluuarvo on tyhjä
Kuvaus	Hyödyntäjä kutsuu operaatiota, kun se tarvitsee koodiarvon perusteella näytettävän arvon yhdestä koodista.
Pakollisuus	Pakollinen kaikissa toteutuksissa
Malli	TQS: CodingSchemeVersion: get_preferred_text
Lisätietoja	TQS-standardi määrittelee myös kontekstista riippuvan paluuarvon (esim. tietyssä käyttötarkoituksessa tai sovelluksessa näytettävä arvo, get_text_for_context). Jos kontekstista riippuvaa esitysmuotoa tarvitaan, myöhemmissä määrittelyissä voidaan lisätä sitä varten operaatioita.

4.2.2 Perustaso (base)

Taulukko 4.2. Perustason (base) operaatiot.

Koodistopalvelussa tuetut koodistot

Operaatio	codeservice: get_supported_codesets
Käyttötarkoitus	Koodistotarjoajan tukemien koodistojen listaus (vaatimus K3.1, K2.2-OID)
Parametrit (in)	(vain palvelun tunniste)
Paluuarvot (out)	codesets: SET (codeset_id:OID, name:ST, version:ST) - Tietojoukko, joka koostuu elementeistä, joissa on koodiston OID-tunniste, koodiston nimi sekä versio.
Poikkeukset	
Kuvaus	Palvelu palauttaa tiedot kaikista koodistoista ja niiden versioista, joita se tukee. Kaikki koodistojen versiot (kokonaiset OID-polut, mukaan lukien versionumero) palautetaan.
Esimerkki	
Pakollisuus	Pakollinen perustason toteutuksissa
Malli	
Lisätietoja	Asiakas voi viitata koodistoon pelkällä koodisto-OID:llä (ilman versiota), jolloin palvelu palauttaa oletusversion (yleensä uusin versio) mukaisia vastauksia. Paikallisten koodistojen (ei-OID-yksilöityjen) käyttö voidaan sallia toteutuksissa. Tällöin niillä on oltava muu yksikäsitteinen tunniste, jonka koodistohyödyntäjä tuntee.

Koodistopalvelussa tuetut palvelutasot

Operaatio	codeservice: get_supported_services
Käyttötarkoitus	Koodistotarjoajan tukemien palvelutasojen listaus (vaatimus K2.8)
Parametrit (in)	(vain palvelun tunniste)
Paluuarvot (out)	services: SET (service_id :ST, name :ST, version :ST) - Tietojoukko, joka koostuu palvelun tunnuksesta, selväkielisestä nimestä ja versiosta
Poikkeukset	
Kuvaus	Palvelu palauttaa tiedot siitä, mitä koodistopalveluita se tukee, tämän dokumentin mukaiset service_id-arvot ovat: (minimitasossa operaatiota ei tueta, palvelu voi palauttaa virheen tai tyhjän joukon get_supported_services-operaatiosta) ”base” – perustaso ”multilingual” – monikieliset koodistot ”freeElements” – vapaa tietosisältö ”advSearch” – edistyneet haut ”hierarchy” – koodistohierarkian käsittely, huom. hierarkkisia koodistoja voidaan käsitellä myös ”ei-hierarkkisinä” ”status” –paikallisten koodien ja tilan huomiointi Palautettava versio viittaa vastaavan määrityksen tunnisteeseen (esim. tämän dokumentin versionumeroon, kuten ”1.0”, tai dokumentin OID-koodiin). Versionumerosta ei voi päätellä, että palvelu tukee edellisiä versioita, vaan kutakin tuettua versiota kohti palvelu palauttaa ”vastausrivin”.
Esimerkki	
Pakollisuus	Pakollinen perustason toteutuksissa
Malli	
Lisätietoja	v0.6 Jatkokehitys: kysyttävä myös koodistokohtaisesti, ks. seur.

Operaatio	codeset: get_supported_codeset_services
Käyttötarkoitus	Yksittäisen koodiston tukemien palvelutasojen listaus kooditarjoajan toteuttamana (vaatimus K2.8, ks. get_supported_services)
Parametrit (in)	(koodiston tunniste)
Paluuarvot (out)	ks. get_supported_services
Poikkeukset	
Kuvaus	ks. get_supported_services
Esimerkki	
Pakollisuus	Pakollinen perustason toteutuksissa
Malli	
Lisätietoja	Lisätty v0.7. Koodistotarjoajan ei tarvitse toteuttaa kaikkia esim. Stakesin koodistopalvelimen tietosisällön mahdollistamia tasoja. Toteutuskohtaisesti on dokumentoitava, mitkä tasot toteutus sisältää kustakin koodistosta.

Koodien haut

Minimitason (haku täsmällisellä koodilla tai selitteellä) lisäksi perustasolla voidaan hakea osittaisella selitteellä tai koodilla. Lisähakuja on määritelty hierarkkisiin luokituksiin ja edistyneet haut tasolle.

Operaatio	codeset: match_concepts_by_string
Käyttötarkoitus	Koodien hakeminen osittaisen selitteen tai koodiarvon avulla (vaatimus K2.6), minimitason lisäksi
Parametrit (in)	text :ST – hakuarvo, mode :ST – hakutapa ("value", "code", "base", tai "freeElements"), oletus "value" how_many :Int – palautettavien maksimimäärä (vapaaehtoinen)
Paluuarvot (out)	codes SET (code_id :ST, value :ST) - Tietojoukko, joka koostuu elementeistä, joissa on koodiarvo sekä selite.
Kuvaus	Palvelu palauttaa ne koodit, joiden selite <u>alkaa</u> hakuarvolla, hakuarvon pienet ja isot kirjaimet vastaavat toisiaan. mode-parametrin merkitys: "value" – haetaan selitteestä (oletus – TQS:n mukainen) "code" – haetaan koodiarvosta "base" – haetaan sekä selitteestä että koodiarvosta "freeElements" – haetaan koodin kunkin sisältöosan alusta (jos toteutettu free-Elements-taso)
Poikkeukset	Paluuarvo on tyhjä joukko, jos hakuarvoa ei löydy minkään selitteen alusta. Jos tuloksia on enemmän kuin on kuvattu tai palvelun tukema maksimimäärä, palvelu palauttaa virhekoodin ja –ilmoitustekstin. Jos mode-parametri on tyhjä tai epäkelpo, palvelu olettaa sen arvoksi "value"
Esimerkki	
Pakollisuus	Pakollinen perustason toteutuksissa. Jos palvelu ei toteuta hakutapoja "value", "code" ja "base", se on mainittava toteutuksen kuvaus-dokumentissa (tuotodokumentaatiossa). "freeElements"-hakutavan toteutus on pakollinen vain jos toteutetaan freeElements-taso.
Malli	TQS: CodingSchemeVersion: match_concepts_by_string: lisätty mode
Lisätietoja	TQS:ssä palautetaan tulosten lisäksi osuman "laatua" kuvaava arvo. Suurin sallittu maksimimäärä jätetään toteutuskohdaisesti määriteltäväksi. Tämä metodi vastaa perushausta myös "vapaa tietosisältö"-tasolla. "Hierarkkiset luokitukset" ja "Edistyneet haut"-tasolla on lisää hakuja, mm. muualta kuin arvojen alusta löytyvien osien haku, nimetyistä sisällön osista, hierarkian osarakenteesta haku jne. Mode-parametri korvattu teknisessä määrittelyssä.

Koodin validointi

Operaatio	codeset: is_valid_concept
Käyttötarkoitus	Koodiarvon olemassaolon tarkistaminen tietyssä koodistossa, sovelluksen kentässä olevan koodin oikeellisuuden varmistaminen (vaatimus K2.9).
Parametrit (in)	code :ST– koodiarvo kokonaisuudessaan
Paluuarvot (out)	valid : BL – true, jos koodiarvo löytyy koodistosta, muuten false
Kuvaus	Käytetään, jos halutaan varmistua siitä, löytyykö koodiarvo koodistosta tai sen versiosta
Poikkeukset	
Esimerkki	
Pakollisuus	Pakollinen perustason toteutuksissa
Malli	TQS: CodingSchemeVersion: is_valid_concept
Lisätietoja	Muutettu merkitys v0.7, ks. myös Status-taso (koodin voimassaolo ja paikallisuus)

Koodistopalvelun hallinnolliset ja yhteystiedot

Operaatio	codeservice: get_info
Käyttötarkoitus	Koodistopalvelun toteutuksen hallinnolliset tiedot (vaatimus K3.2)
Parametrit (in)	(palvelun tunniste)
Paluuarvot (out)	info:LIST (info_element:ST) – vapaamuotoinen merkkijonolista
Kuvaus	palautettavasta listasta voi selvittää mm. tuote ja versio, toteuttaja sekä koodisto-tarjoajan toteuttajan yhteystiedot
Poikkeukset	
Esimerkki	
Pakollisuus	Toteutettava perustasolla, mutta voi palauttaa myös tyhjän
Malli	
Lisätietoja	siirretty perustasolle v0.9

Koodiston hallinnolliset tiedot

Operaatio	codeset: get_codeset_info
Käyttötarkoitus	Koodiston hallinnolliset tiedot (vaatimus K2.4)
Parametrit (in)	(koodiston tunniste)
Paluuarvot (out)	info:LIST (info_element:ST) – vapaamuotoinen merkkijonolista
Kuvaus	palautettavasta listasta voi selvittää mm. koodiston ylläpitäjä, yhteystiedot, osoite johon lähettää kehitysideat, päivitysväli jne.
Poikkeukset	
Esimerkki	
Pakollisuus	Toteutettava perustasolla, mutta voi palauttaa myös tyhjän
Malli	
Lisätietoja	Huom. koodiston versiotieto on pakollinen (määritelty perustasolla omassa operaatiossaan) get_codeset_info siirretty perustasolle v0.9

4.2.3 Monikieliset koodistot (multilingual)

Monikieliset koodistot-toteutuksissa palvelu voi tarjota koodiarvoja vastaavia selitteitä useilla kielillä niissä koodistoissa, joissa on usean kielisiä selitteitä.

Taulukko 4.3. Monikieliset koodisto (multilingual)-tason operaatiot.

Operaatio	codeset: list_languages
Käyttötarkoitus	Tiettyssä koodistossa tuettujen kielten listaus (vaatimus K2.12)
Parametrit (in)	(koodiston tunniste)
Paluuarvot (out)	languages: LIST (language_code:ST, language:ST) - Lista, joka koostuu elementeistä, joissa on kielikoodi (SFS-ISO 639) sekä kielen nimi.
Kuvaus	Koodistokohtaisten kielten selvittäminen
Poikkeukset	
Esimerkki	paluuarvo esim. ”fi”, ”suomi”
Pakollisuus	Pakollinen, jos toteutetaan ”monikieliset koodistot”-taso
Malli	
Lisätietoja	Kielen kysely voidaan tarvittaessa määritellä myös koodi-rajapintaan myöhemmissä versioissa (jos eri koodeissa eri tuetut kielet). Kielten koodisto tarkennettu v0.9.

Operaatio	code: get_lang_text
Käyttötarkoitus	Halutun kielisen selitteen hakeminen sovellukseen tietyllä koodilla (vaatimus K4.4)
Parametrit (in)	ks. <code>get_preferred_text</code> , lang_code :ST – kielikoodi (SFS-ISO 639)
Paluuarvot (out)	value :ST - Selite (näytettävä arvo), halutulla kielellä, jos useita kieliä käytössä
Kuvaus	Koodihyödyntäjä saa kooditarjoajalta yhden koodin halutun kielisen selitteen
Poikkeukset	Jos annetulla koodiarvolla ei löydy koodia, paluuarvo on tyhjä. Jos kieltä ei ole tuettu koodistossa tai koodissa, palautetaan virhe.
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”monikieliset koodistot”-taso
Malli	
Lisätietoja	

Vastaavasti kuin `get_lang_text`, ”monikieliset koodistot”-tasolla toteutetaan operaatiot `get_concepts_by_text_lang`, `match_concepts_by_string_lang` ja `list_concepts_lang`, joissa on parametrina operaatioissa muuten kuvattujen lisäksi `lang_code`:ST-kielikoodi, ja joissa sekä in että out-parametreissa käytetään halutun kielisiä selitteitä.

4.2.4 Vapaa tietosisältö (freeElements)

Vapaa tietosisältö- toteutuksissa palvelu tarjoaa koodiarvojen ja selitteiden lisäksi muita tietokenttiä koodistojen sisällä. Mm. multilingual-, hierarchical- ja status-tasojen toiminnot voidaan johtaa freeElements-tason toiminnoista, jos niihin tarvittavia tietoja saadaan freeElements-tasolla.

Taulukko 4.4. Vapaa tietosisältö (freeElements)-tason operaatiot.

Koodien listaus

Operaatio	codeset: list_concepts_and_values
Käyttötarkoitus	Koodien listaus halutulla tietosisällöllä (ei-hierarkkisessa koodistossa)
Parametrit (in)	how_many :Int – kerralla palautettavien maksimimäärä (vapaaehtoinen), from :ST – ensimmäisen haettavan koodin koodiarvo (tunniste), tyhjä jos haetaan alusta lähtien elements : LIST (element :ST) – pyydettyjen tietokenttien nimet sortBy : ST (vapaaehtoinen) – minkä kentän mukaan lajiteltuna tulokset palautetaan
Paluuarvot (out)	more : ST – ensimmäisen vielä jäljellä olevan koodin koodiarvo (tunniste). Jos kaikki palautettu, tyhjä merkkijono. codes : LIST (code_id :ST, values :LIST(value :ST)) - Lista, joka koostuu elementeistä, joissa on koodiarvo sekä pyydytyssä järjestyksessä (tai XML-tekniikalla nimettynä) lista pyydettyjen kenttien arvoista kyseisessä koodissa.
Kuvaus	ks. <code>list_concepts</code> . Tietoelementtien nimeämisessä suositellaan kans. koodistopalvelimen mukaisia elementtien nimiä (ks. liite 1). SortBy-parametrissa voidaan antaa haluttu lajittelukenttä, jonka mukaisessa nousevassa järjestyksessä tulokset halutaan (oletusarvo on koodiarvon mukainen järjestys).
Poikkeukset	Jos koodiston tietokentissä ei ole jotain nimetyistä kentistä, palvelu palauttaa virheen. Jos palvelu ei tue lajittelua pyydetyn kentän mukaan, palvelu palauttaa virheen. Koodistohyödyntäjä ei saa vaihtaa sortBy-parametria alkuperäiseen hakuun verrattuna, jos tehdään jatkohaku.
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”vapaa tietosisältö”-taso.

Malli	
Lisätietoja	ks. list_concepts. TQS:ssä perushakuoperaatioihin ei määritelty muuta tietosisältöä kuin koodiarvo ja selite, mutta laajemmalle tai valitulle sisällölle myös listauksessa on tarvetta. Lajitteluun (lisätty v0.9) voidaan myöhemmissä versioissa lisätä nouseva/laskeva järjestys.

Yhden koodin koko sisällön hakeminen (huom. code-rajapinta)

Operaatio	code: get_values
Käyttötarkoitus	Kaikkien koodin tietojen hakeminen sovellukseen tietyllä koodilla (vaatimus K4.2)
Parametrit (in)	(koodin yksilöinti)
Paluuarvot (out)	values:SET (element:ST, value:ST) – Lista, joka sisältää yhdestä koodista jokaisen elementin nimen ja arvon. Jos koodi sisältää elementin muttei arvoa, palautetaan element ja tyhjä value.
Poikkeukset	Jos annetulla koodiarvolla ei löydy koodia, paluuarvo on tyhjä.
Kuvaus	Hyödyntäjä kutsuu operaatiota, kun se tarvitsee koodiarvon perusteella kaiken sisällön yhdestä koodista.
Pakollisuus	Pakollinen, jos toteutetaan ”vapaa tietosisältö”-taso
Malli	TQS: CodingSchemeVersion: get_preferred_text
Lisätietoja	TQS-standardi määrittelee myös kontekstista riippuvan paluuarvon (esim. tietyssä käyttötarkoituksessa tai sovelluksessa näytettävä arvo, get_text_for_context). Jos kontekstista riippuvaa esitysmuotoa tarvitaan, myöhemmissä määrittelyissä voidaan lisätä sitä varten operaatioita.

Yhden koodin haluttujen sisältöelementtien hakeminen (code-rajapinta)

Operaatio	code: get_value
Käyttötarkoitus	Yhden koodin halutun sisältöelementin hakeminen sovellukseen (vaatimus K4.3)
Parametrit (in)	(koodin yksilöinti), element:ST – halutun sisältöelementin nimi
Paluuarvot (out)	value:ST Selite (näytettävä arvo), palvelussa määritellyllä oletuskielellä, jos useita kieliä käytössä
Poikkeukset	Jos annetulla koodiarvolla ei löydy koodia, tai koodissa ei ole nimettyä elementtiä, paluuarvo on tyhjä
Kuvaus	Hyödyntäjä kutsuu operaatiota, kun se tarvitsee koodiarvon perusteella jonkin sisältökentän arvon yhdestä koodista.
Pakollisuus	Pakollinen, jos toteutetaan ”vapaa tietosisältö”-taso
Malli	
Lisätietoja	

Nimettyjen koodien haluttujen sisältöelementtien hakeminen

Operaatio	codeset: get_concepts_and_values
Käyttötarkoitus	Saman koodiston usean yksilöidyn koodin listaus halutulla tietosisällöllä
Parametrit (in)	code_ids: LIST (code_id: ST) – lista pyydettyjen koodien tunnisteita elements: LIST (element: ST) – pyydettyjen tietokenttien nimet
Paluuarvot (out)	codes: LIST (code_id: ST, values: LIST(value: ST)) - Lista, joka koostuu elementeistä, joissa on koodiarvo sekä pyydettyssä järjestyksessä (tai XML-tekniikalla nimettynä) lista pyydettyjen kenttien arvoista kyseisessä koodissa.
Kuvaus	ks. list_concepts_and_values. Esim. haun tuloksena saaduista koodien tunnistuksista voidaan tämän operaation avulla hakea halutut kentät näytettäväksi (ilman toistuvia hakuja tai koodikohtaista tietojen hakua).
Poikkeukset	Jos koodiston tietokentissä ei ole jotain nimetyistä kentistä, palvelu palauttaa virheen. Toteutuskohtaisesti voidaan määritellä maksimimäärä sallittuja code_id-arvoja.
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”vapaa tietosisältö”-taso
Malli	
Lisätietoja	ks. list_concepts_and_values.

Koodiston sisältämien koodien tietokenttien kysely

Operaatio	codeset: get_elements
Käyttötarkoitus	Koodiston sisältämien sisältöelementtien nimien hakeminen (vaatimus K2.1)
Parametrit (in)	(koodiston yksilöinti)
Paluuarvot (out)	elements: LIST (element_name: ST), Lista tietokenttien nimistä
Poikkeukset	
Kuvaus	Hyödyntäjä kutsuu operaatiota, kun se tarvitsee tiedon siitä, mitä sisältöelementtejä tietty koodisto kaiken kaikkiaan sisältää.
Pakollisuus	Pakollinen, jos toteutetaan ”vapaa tietosisältö”-taso
Malli	
Lisätietoja	Teknisessä määrittelyksessä voidaan tarkentaa, että myös elementin tyyppin palauttaminen vaaditaan.

Myös match_concepts_by_value (advSearch-tasolla) liittyy vapaa tietosisältö-tasoon.

4.2.5 Edistyneet haut (advSearch)

Taulukko 4.5. Edistyneet haut (advSearch)-tason operaatiot.

Koodien haut

Operaatio	codeset: match_concepts_by_substring
Käyttötarkoitus	Koodien hakeminen mistä tahansa koodien sisällöstä löytyvän merkkijonon avulla (vaatimus K2.6, laajennettuna)
Parametrit (in)	texts: LIST(text: ST) – lista hakuarvoja, how_many: Int – palautettavien maksimimäärä (vapaaehtoinen)
Paluuarvot (out)	codes: LIST (code_id: ST, value: ST) - Lista, joka koostuu elementeistä, joissa on koodiarvo sekä selite.
Kuvaus	Palvelu palauttaa ne koodit, joiden selitteestä, koodiaresta tai mistä tahansa sisällöstä (esim. erikieliset selitteet) löytyy hakuarvoa vastaava osa, hakuarvon pienet ja isot kirjaimet vastaavat toisiaan. Arvot palautetaan palvelun määrittelemässä ”paremmuusjärjestyksessä”, jonka muodostukseen voidaan hyödyntää hakuarvojen järjestystä ja niiden esiintymistä löydettyissä osumissa.

Poikkeukset	Paluarvo on tyhjä joukko, jos hakuarvoa ei löydy minkään koodin sisällöstä. Jos tuloksia on enemmän kuin kuvattu tai palvelun tukema maksimimäärä, palvelu palauttaa virheen.
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”edistyneet haut”-taso
Malli	TQS: CodingSchemeVersion: match_concepts_by_string
Lisätietoja	ks. match_concepts_by_string. Myöhemmissä versioissa osumasta voidaan palauttaa myös lisätietoa, esim. minkä elementin sisällä se on.

Operaatio	codeset: match_concepts_by_keywords
Käyttötarkoitus	Koodien hakeminen avainsanojen avulla (vaatimus K2.11)
Parametrit (in)	keywords LIST (keyword :ST) – lista avainsanoja, how_many :Int – palautettavien maksimimäärä (vapaaehtoinen)
Paluarvot (out)	codes LIST (code_id :ST, value :ST) - Tietojoukko, joka koostuu elementeistä, joissa on koodiarvo sekä selite.
Kuvaus	Palvelu palauttaa ne koodit, joille määritellyistä avainsanoista löytyy hakuarvoa vastaava osa, muuten ks. match_concepts_by_substring.
Poikkeukset	ks. match_concepts_by_substring.
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”edistyneet haut”-taso
Malli	TQS: CodingSchemeVersion: match_concepts_by_keywords
Lisätietoja	

Operaatio	codeset: match_concepts_by_value
Käyttötarkoitus	Koodien hakeminen nimettyjen elementtien sisällön perusteella (vaatimus K2.6, laajennettuna)
Parametrit (in)	keys :LIST(element :ST, text :ST) – lista hakuarvoja, how_many :Int – palautettavien maksimimäärä (vapaaehtoinen)
Paluarvot (out)	codes :LIST (code_id :ST, value :ST) - Lista, joka koostuu elementeistä, joissa on koodiarvo sekä selite.
Kuvaus	Palvelu palauttaa ne koodit, joiden nimettyjen elementtien sisällöstä löytyy vastaavia hakuarvoja vastaavat osat. Muuten ks. match_concepts_by_substring.
Poikkeukset	ks. match_concepts_by_substring.
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”edistyneet haut”- ja ”vapaa tietosisältö”-tasot
Malli	
Lisätietoja	Myöhemmissä voidaan määritellä palauttaviksi myös niiden elementtien sisältö, joita haussa on käytetty. Tähän voidaan käyttää myös get_concepts_and_values –operaatiota (freeElements)

4.2.6 Hierarkkisten koodistojen käsittely (hierarchical)

Hierarkkisten koodistojen käsittely- tason operaatiot vastaavat vaatimukseen K2.10. Tällöin palvelu toteuttaa (myös) operaatiot:

- codeset: get_concepts_by_text_hier (lapisolmuista käsiteellä haku)
- codeset: match_concepts_by_string_hier (lapisolmuista käsitteen alulla haku)
- codeset: match_concepts_by_substring_hier (jos toteutettu myös edistyneet haut)

- codeset: match_concepts_by_keywords_hier (jos toteutettu myös edistyneet haut)
- codeset: list_concepts_hier (lapisolmujen listaus)
- codeset: list_concepts_and_values_hier (lapisolmujen listaus)

Nämä operaatiot ovat kuten edellä kuvatut muut operaatiot (ilman hier-päätettä), mutta:

- Operaation nimen perään lisätään osuus _hier
- In-parametrina on myös hier_root:ST, jolla tunnistetaan alihierarkian juuri
- Haku- ja listausoperaatioiden paluuarvoissa (codeset-rajapinnan tasolla) palautetaan myös vastausten hierarkian taso hier_level:Int (ylin taso=0)
- Hakuoperaatiot palauttavat vain annetun juuren alla olevan tason tulokset. Hakuoperaatioita voidaan myös kutsua näille tuloksille rekursiivisesti, tai käyttää hakuja, joissa hierarkiaa ei oteta huomioon.

Teknisissä määrittelyissä on mahdollista lisätä perusoperaatioihin hierarkian käsittelyyn tarvittavat osat (ja palauttaa virhearvoja, jos palvelu ei tue niitä silloin, kun koodistohyödyntäjä yrittää niitä käyttää).

Lisäksi koodisto-rajapinnassa on toteutettu operaatio:

Taulukko 4.6. Hierarkkiset koodistot (hierarchy)-tason lisäoperaatiot.

Operaatio	codeset: hier_depth
Käyttötarkoitus	Koodistohierarkian syvyyden kysely
Parametrit (in)	hier_root:ST – haettavan koodin koodiarvo (tunniste), tyhjä jos haetaan juuresta
Paluuarvot (out)	depth:Int , koodistohierarkian syvyys
Kuvaus	Palauttaa juuresta tai halutusta solmusta laskien hierarkian syvyyden. 0=ei alitasoja, yksitasoisen koodiston juuressa (ilman parametria) 1 (v0.9)
Poikkeukset	
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”hierarkkisten koodien käsittely”-taso
Malli	
Lisätietoja	TQS:n mallissa hierarkiat käsitellään muiden suhteiden ja assosiaatioiden yhteydessä. Huom. hierarkian taso elementeille (eri asia) palautetaan hakutuloksissa. Koodi-rajapintaan lisätty tason kysely v0.8.

Ja code-rajapinnassa operaatio:

Operaatio	code: hier_parent
Käyttötarkoitus	Hierarkiassa ylemmän tason elementin kysely
Parametrit (in)	(vain koodiarvo –tunniste)
Paluuarvot (out)	parent:ST , ylemmän tason elementin (vanhempi-elementti) koodiarvo (tunniste)
Kuvaus	Palauttaa halutun elementin vanhemman tunniste
Poikkeukset	Palauttaa tyhjän arvon, jos ylempi taso on juuri (jolla ei omaa koodia)
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”hierarkkisten koodien käsittely”-taso
Malli	
Lisätietoja	

Lisäksi code-rajapinnassa on tuettava yksittäisen koodin hierarkiatason kyselyä, tukemalla `code:get_value(hierarchyLevel)` –operaatiota (operaatio on toteutettava myös, jos tuetaan ”Hierarchy” tasoa, muttei ”freeElements”-tasoa. Operaatio palauttaa ko. koodin hierarkiatason (0=ylin).

4.2.7 Paikalliset koodit ja koodien tila (status)

Status-taso vastaa vaatimukseen K2.15. Oletusarvoisesti (ellei tätä tasoa ole toteutettu), palautetaan kaikki koodit koodistosta niitä kysyttäessä, välittämättä siitä, ovatko ne paikallisia tai onko niiden tila poistettu, ehdotettu tai käytössä. Kun tämä taso toteutetaan, voidaan esim. hakutuloksille tai koodeille tehdä lisäkäsittelyä paikallisuus- ja tilatietojen perusteella.

Taulukko 4.7. Paikalliset koodit ja koodien tila (status)-tason operaatiot.

Operaatio	code: get_status
Käyttötarkoitus	Koodin tilan kysely (käytössä, ehdotettu, poistettu)
Parametrit (in)	(vain koodiarvo –tunniste)
Paluuarvot (out)	status:ST , tilatieto, jokin seuraavista ”active” – käytössä ”work” – ehdotettu ”delete” – poistettu
Kuvaus	Palauttaa kysytyn koodin tilatiedon, oletuksena active
Poikkeukset	
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”paikalliset koodit ja tila”-taso
Malli	Koodistopalvelu, status-tieto, voidaan mukauttaa TQS:n <code>is_valid_concept</code> – mukaiseksi
Lisätietoja	Lisätty v0.7, voidaan käyttää myös hakuoperaatioiden yhteydessä, ellei haluta määritellä (esim. teknisissä määrittelyksessä) erikseen ”tilatietoisia” hakuja

Operaatio	code: is_local_concept
Käyttötarkoitus	Koodin paikallisuuden kysely
Parametrit (in)	(vain koodiarvo –tunniste)
Paluuarvot (out)	local:BL , true, mikäli koodi ei ole alkuperäisessä (esim. kansallisessa) koodistossa vaan on lisätty paikallisesti
Kuvaus	Palauttaa kysytyn koodin paikallisuustiedon, oletuksena false
Poikkeukset	
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”paikalliset koodit ja koodien tila”-taso
Malli	Koodistopalvelu, local-tieto (ST), mukailtu TQS:n <code>is_valid_concept</code>
Lisätietoja	Lisätty v0.7, määrittely ei ota kantaa, onko koodi lisätty palveluun juuri sillä tasolla, jolla palvelu toimii vai jollain ylemmällä tasolla (esim. lisäys voi tapahtua alueellisella tasolla ja palvelu sijaita organisaatiotasolla, jolloin <code>get_local</code> palauttaa true).

Operaatio	codeset: list_valid_concepts
Käyttötarkoitus	Käytössä (active) olevien koodien listaus
Parametrit (in)	how_many :Int – kerralla palautettavien maksimimäärä (vapaaehtoinen), from :ST – ensimmäisen haettavan koodin koodiarvo (tunniste), tyhjä jos haetaan alusta lähtien
Paluuarvot (out)	more : ST – ensimmäisen vielä jäljellä olevan (palauttamattoman) koodin koodiarvo (tunniste). Jos kaikki palautettu, tyhjä merkkijono. codes LIST (code:ST, value:ST) - Tietojoukko, joka koostuu elementeistä, joissa on koodiarvo sekä selite
Kuvaus	Palvelu palauttaa listan koodiston käytössä olevista koodeista koodiarvon määrittelemässä järjestyksessä. Peräkkäisillä hauilla voidaan jatkaa listausta ensimmäisestä vielä palauttamattomasta koodista. ks. list_concepts
Poikkeukset	
Esimerkki	
Pakollisuus	Pakollinen, jos toteutetaan ”paikalliset koodit ja koodien tila”-taso
Malli	Koodistopalvelu, status-tieto (ST), TQS: CodingSchemeVersion: list_concepts
Lisätietoja	ks. get_status

4.3 Muut tekniikkariippumattomat määritykset

4.3.1 Yleiset virhetilanteet

Tarkka virheiden palautusmekanismi on tekniikkakohtainen. http- ja XML-tekniikoille virheet on tarkennettu luvussa 5.9. Kohdassa poikkeukset teksti ”palvelu palauttaa virheen” tarkoittaa sitä, että toiminta tarkennetaan teknisissä määrittelyissä. Tyypillisesti virhe palauttaa esim. virhekoodin ja mahdollisen virheilmoituksen, tai määritellyn virhettä tarkoittavan paluuarvon.

Ohjeelliset virhekoodit tässä dokumenteissa määritellyille useisiin operaatioihin liittyville virheille:

Koodi Ilmoitus esim.

- 1 Operaatiota ei ole toteutettu
- 2 Pakollinen tieto puuttuu (parametrin nimi)
- 3 Liian monta pyydettyä koodia
- 4 Pyydettyä kieltä ei ole koodistossa tai koodipalvelussa
- 5 Nimetty tieto ei sisälly koodistoon

4.3.2 Koodistohyödyntäjä-esimerkkejä

Koodistohyödyntäjä voi hyödyntää koodistotarjoajan palveluita monella tavalla:

- Käyttäjän tekemissä hauissa, koodistojen selaamisessa, kenttien arvojen varmistamisessa, ja muissa koodistotarjoajan tarjoamia palveluita vastaavissa tilanteissa voidaan hyödyntää suoraan jaettuja koodistoja koodistotarjoajalta. Koodistoja tai haku- tai vastaavia operaatioita ei tarvitse kopioida tai toteuttaa koodistohyödyntäjän ohjelmistossa (joskin myös koodiston kopiointi rajapintojen avulla on mahdollista).
- Jos koodistohyödyntäjä tallettaa omaan tietovarastoonsa viitteen koodistopalvelimen sisältämään koodiarvoon, talletettava tieto voi koostua esim.:

- Vähimmillään koodiarvosta, jos käytettävä koodisto tunnetaan ja on vakio. Tällöin koodistohyödyntäjä voi käyttää koodistopalvelua myös käyttäjälle näytettävän arvon hakuun (get_preferred_text).
- Koodiarvosta ja näytettävästä arvosta (vast. HL7 CS-tietotyyppi). Tällöin koodistohyödyntäjä voi näyttää sekä koodiarvon (jos se näytetään) että selitteen omasta tietovarastostaan, ja koodistotarjoajan palvelua ei käytetä näytettävän arvon hakuun vaan muihin, esim. haku-, selaus- ym. toimintoihin.
- Myös koodiarvosta, koodiston tunnisteesta (esim. versio) ja näytettävästä arvosta (vast. HL7 CV-tietotyyppi). Tämä mahdollistaa viittaukset ja varmistukset koodistoista, joiden versiot muuttuvat usein.
- Koodistotarjoajan operaatioista saatua tietoa voidaan luonnollisesti tallettaa vapaasti koodistohyödyntäjän käyttötärpeiden mukaisesti (vaikka kokonainen koodisto, joskin kopiointiin voidaan käyttää myös tietokannan tasolla integraatiota tai Stakesin koodistopalvelimen XML-siirtomuotoa). Uusimpien versioiden ja muutosten käyttöönsaanti helpottuu kuitenkin, mitä vähemmän tietosisältöä koodistotarjoajasta kopioidaan koodistohyödyntäjään.

Ks. myös kuvat 1.1 ja 4.1 ja esimerkit luvussa 4.4.

4.4 Integrintiteknikoiden vaatimukset

Luvussa 4 määritellyistä rajapinnoista voidaan tuottaa teknisiä määrittelyitä eri integrintiteknikoille (esim. paikallisia kirjastorajapintoja, verkon yli käytettäviä hajautettuja oliopalveluita tai web-sovelluspalveluita). Kerralla tekniseen määrittelyyn voidaan valita tästä määrittäyksestä haluttaessa vain osa, esim. minimi- ja perustaso.

Kussakin teknisissä määrittelyissä tulisi mm. määritellä:

- mikä on tarkka mekanismi minimitason operaatioille koodiston ja koodin käyttöön saamiseksi,
- käytetäänkö XML:ää vai esim. ohjelmointikielten tai liittymämäärittelyjen tietorakenteita, erotetaan tietosisältö XML-merkkauksella toiminnoista (aina tai esim. vain freeElements-tasolla),
- kuinka määritellyt toiminnot (operaatiot) ja niiden tarvitsema ja palauttama tietosisältö ja tietotyypit teknisesti määritellään (tietotyypeissä voidaan perustellusti poiketa tässä kuvatuista),
- käytetäänkö tietosisällön merkkauksessa samaa XML-muotoa tai sen osajoukkoa, jota käytetään Stakesin Koodistopalvelu-määrittelyissä (Koodistopalvelin 2003: SOAP Body-osion sisällä oleva XML-määrittely),
- kuinka koodistohyödyntäjä voi paikallistaa ja saada käyttöönsä koodistotarjoajan tarjoaman palvelun,
- miten vapaaehtoisiksi merkityt parametrit käsitellään, esim. lähetetäänkö parametria lainkaan vai onko niiden tilalla tyhjä arvo tai puuttuvaa tarkoittava arvo, ja kuinka tyhjät arvot teknisesti toteutetaan.

Integroinnin teknistä määrittelyä ja toteutusta varten on hyödynnetty luvussa 5 samaa XML- ja http-pohjaista toteutustapaa, jota on käytetty mm. PlugIT-projektin muissa Ydinpalveluissa. Tämä tekniikka vastaa esitettyihin avoimuusvaatimuksiin ja on suhteellisen kevyt toteuttaa sovittimena ole-massa oleviin sovelluksiin. Useissa saaduissa kommentteissa tämä tekniikka on nähty ensimmäisenä määrittelytarpeena. Myös muita tekniikoita, kuten web-sovelluspalveluiden SOAP- ja WSDL-määrittelyjä sekä sovellusten toteutuksissa käytettäviä (esim. Java- tai COM/dll-) tekniikoita voitaisiin valita teknisiin määrittelyihin. Eri rajapintateknikoiden arviointia on mm. raportissa (Mykkänen ym. 2004a).

Koodistomäärittelysten tekninen määrittely ja tarkennus eri tekniikoilla mahdollistaa esim. seuraavanlaiset palvelutoteutukset tämän määrittelyksen pohjalta. Teknisissä määrittelyissä voidaan yh-

distää eri rajapintoja, esim. koodi-rajapinnan operaatiot voi tarjota myös koodisto-rajapinta. (Seuraavassa esimerkkinä yhden koodin näytettävän arvon haku ICD-10-koodistosta, perustaso):

Esimerkki olio-ohjelmointikielellä (codeset ja code-rajapintojen käyttö):

```
var Service: ICodeService;  
    ICDCodeSet: ICodeSet;  
    ICDCode: ICode;  
..  
ICDCodeSet:=Service.Get_CodeSet('1.2.246.537.6.1.1996');  
ICDCode:=ICDCodeSet.Get_Code('B33.2');  
Edit1.Text := ICDCode.get_preferred_text;  
// 'Viruksen aiheuttama sydäntulehdus'
```

Esimerkki http-pohjaisena palveluna (operaatio, codeset ja code parametreina, eli rajapinnat yhdistetty codeset-rajapinnan tasolla, huom. eri käytäntö (GET) kuin luvun 5 teknisessä määrittelyssä (POST)):

```
http://www.uku.fi/codeservice.jsp?op=get_preferred_text&codeset=1%2E2%2E24  
6%2E537%2E6%2E1%2E1996&code=B33%2E2
```

ja vastaus esim. XML-dokumenttina:

```
<?xml version='1.0' encoding='iso-8859-1'?>  
<code lang="fi" value="Viruksen aiheuttama sydäntulehdus"/>
```

Operaatioita ja parametreja (ks. liite 3) voidaan toteutustekniikoiden sallimilla tavoilla myös yhdistää eri tasoilla teknisissä määrittelyissä, jolloin voidaan korvata osa parametreista oletusarvoilla, ja näin operaatioiden lukumäärä pienenee. Esimerkiksi koodien listausoperaatioon voidaan yhdistää eri tasojen vaatimia parametreja. Tällöin esim.

```
codeset.list_concepts (how_many, from, elements, lang, hier_root, status)
```

voisi palauttaa koodistosta elements-parametrin mukaiset sisältöelementit, selitteen lang-parametrin mukaisella kielellä, vain hier-parametrin osoittaman solmun alla hierarkiassa löytyvät solmut, ja status-parametrin kertomana vain sellaiset jotka on merkitty käytössä oleviksi. Jos jokin toteutus ei tukisi kaikkia vaadittuja tasoja ja ei-tuettuun tasoon kuuluva parametri poikkeaisi oletusarvosta, palvelu antaisi virheilmoituksen.

4.5 Toteutuksiin kohdistuvat vaatimukset

4.5.1 Toteutuskohtaisiksi jätettävät seikat

Tässä määrittelyssä ei ole määritelty:

- Mitkä koodistot palvelu tarjoaa (esim. ICD-10, organisaatiokoodisto jne.). Stakesin koodistopalvelimelle suunnitelluista koodistoista on lista luvussa 2.5 (ks. myös Mykkänen ym. 2004b, jossa on luettelo ja linkit keskeisiin koodistostandardeihin).
- Vaatiiko koodistotarjoaja koodistohyödyntäjä-sovelluksen tai käyttäjän yksilöintiä ja kuinka tämä yksilöinti ja sen varmistaminen tapahtuu. Tarvittaessa tunnistus ja yksilöinti voidaan toteuttaa esim. PlugIT-projektin käyttäjä-rajapinnan määritysten mukaisesti (Sormunen ym. 2004b).
- Vastuita siinä tapauksessa, että koodistotarjoaja on esim. teknisen vian vuoksi saavuttamattomissa. Tämä ja vastaavat tilanteet sovitaan tuotekohtaisesti.

- Jos samassa koodistossa yhdellä (varsinaisella) koodiarvolla on useita selitteitä, on dokumentoitava koodistokohtaisesti, käytetäänkö yksilöivänä tunnisteena (koodiarvo) jotain muuta tietoa kuin varsinaisen koodiarvo (esim. ”rivinumero”) (v0.9). Tunnisteeksi suositellaan samaa tunnistetta kuin Stakesin koodistopalvelu-määrittelyissä. Tällöin on myös kuvattava, vastaako esim. peruslistaus-operaatioiden ”selite/value” koodiarvoa vai selitettä, ja käytetäänkö esim. freeElements-tason operaatioita varsinaisen selitteen tai koodiarvon hakuun. Suositus on käyttää tällöin value-arvona ”koodiarvoa” ja noutaa selitteet tarvittaessa esim. freeElements-mekanismeilla.
- Näitä seikkoja tulisi kuvata tapaus- ja toteutuskohtaisesti mm. tuotedokumentaatioissa ja tarjouspyynnöissä.

Toteutuksessa tulisi dokumentoida yleisten toteutuskohtaisten käytäntöjen dokumentoinnin (Mykänen ym. 2004c) lisäksi mm.:

- mitkä koodistot ja niiden versiot palvelu tarjoaa,
- mitkä tasot määrittelyksestä on toteutettu palveluun ja eri koodistoihin (esim. ”minimi”, ”perus”, ”vapaa tietosisältö” ja ”paikalliset koodit ja koodien tila”),
- mikä on hakuoperaatioissa kerralla palautettavien vastausten määrän suurin sallittu arvo ja oletusarvo (jos hyödyntäjä ei määrittele how_many-parametreja),
- mikä on suurin sallittu määrä kerralla pyydettyjä nimettyjä koodiarvoja (freeElements-taso),
- mainittava, mikäli vain jotkin perustason match_concepts_by_string –”value”, ”code” ja ”base”-hakutavoista on toteutettu,
- mikä on oletuskieli monikielisissä koodistoissa,
- mitkä ovat freeElements-tasolla tuetut sisältöelementit kussakin koodistossa,
- tukeeko palvelu koodien listauksessa tulosten lajittelua minkä tahansa kentän mukaan, tai minkä kenttien mukaan lajittelua palvelu tukee (v0.9) (freeElements-taso),
- asetetaanko koodistohyödyntäjälle lisärajoitteita tai –vaatimuksia esim. koodistojen massiivisen toistuvan listauksen tai muiden koodistotarjoajaa runsaasti kuormittavien operaatioiden suhteen.

4.5.2 Määrittelyn mukaisuus

Vaatimusmäärittelyjen kohdalla ne vaatimukset, jotka merkitään pakollisiksi (esim. kriittisen prioriteetin vaatimukset tai valitut tekniikkariippumattoman määrittelyn palvelutasot) voidaan asettaa hyväksymistestauksen kriteereiksi. Tällä tasolla palvelun toteutus on vaatimusten mukainen, jos sen toiminta ja palauttavat arvot ovat määriteltyjen integrointivaatimusten mukaisia. Nämä seikat voidaan todeta testausraportista, joka on tuotettu joko kehitys- tai käyttöympäristössä. Hyväksymistestauksen suorittaminen on suositeltavaa käyttöympäristössä.

Tekniikkariippumattomien määrittelyjen kohdalla on kerrottu, mitkä operaatiot ja tietoelementit kuuluvat minimitasoon ja muihin tasoihin, ja mainittu, mitkä seikat ovat palvelun tarjoajan ja palvelun käyttäjän vastuulla. Näitä määriteltyjä tasoja (toimintokokonaisuuksia) tässä versiossa ovat:

- minimitaso,
- perustaso,
- monikieliset koodistot,
- vapaa tietosisältö,
- edistyneet haut,
- koodistohierarkian käsittely ja
- paikalliset koodit ja koodien tila.

Toteutus on tämän dokumentin mukainen, jos se toteuttaa minimitason. Toteutukset voivat toteuttaa suositeltavien minimi- ja perustason lisäksi useita muita tasoja (toisistaan riippumatta). Mm. moni-

kieliset koodistot, koodistohierarkian käsittely, paikalliset koodit ja koodien tila-tasot on nähty yleisimpinä koodistotarjoajaan kohdistuvina tarpeina saaduissa kommenteissa. Tarjouspyynnöissä ja integrointiratkaisuissa voidaan vaatia tiettyjen tai kaikkien tasojen toteuttamista. On huomioitava että avoimia ja ”suoraan yhteen toimivia” rajapintoja haluttaessa tarjouspyynnöt ja integrointiratkaisut kannattaa tehdä tarkkojen teknisten liittymämäärittelyjen (luku 5) pohjalta tämän luvun tekniikkariippumattomien määritysten sijaan.

Toteutukset, jotka noudattavat kuvattuja vaatimuksia, määriteltyjä sovellusten vastuita ja luvussa 4 kuvattuja rajapintoja, ovat ”Koodistorajapinnat: Tekniikkariippumattomat liittymämäärittelyt, versio 1” määrittelyn mukaisia. Tällä tasolla palvelun toteutuksen määritysten mukaisuus voidaan todeta testausraportilla, joka sisältää palvelun toteutuksen palauttavat arvot ja toiminnan kuvauksen yllä kuvatuilla ehdoilla ja lisäksi joillakin virheellisillä ja puuttuvilla syötteillä. Palveluja käyttävän sovelluksen määritysten mukaisuus voidaan todeta testausraportilla, jossa palvelun onnistuneiden kutsujen (yllä olevilla edellytyksillä) lisäksi selviää palvelua käyttävän sovelluksen toiminta palvelun palauttaessa virhetilanteita tai ollessa saavuttamattomissa. Määrittelyjen mukaisuuden testausta on käsitelty raportissa (Mykkänen ym. 2004d).

4.6 Määrittelyn käyttö ja jatkokehitys

Luvun 3 Integrointivaatimukset ja luvun 4 Tekniikkariippumattomat määrittelyt ovat olleet pohjana luvun 5 teknisille määrittelyille. Niitä voidaan hyödyntää myös irrallaan teknisistä määrittelyistä asetettaessa tavoitteita ja vaatimuksia koodistoja käsitteleville ohjelmistoille ja rajapinnoille tai kehitettäessä asiaan liittyviä muita määrittelyjä.

Määrittelyistä pois jätetyt osiot ja jatkokehitys

Vaatimusmäärittelyissä (luku 3) joukko vaatimuksia on siirretty jatkokehitysideoiksi.

Määrittelyistä pois jätetyt osiot ja Tekniikkariippumattoman määrittelyn jatkokehitysideat:

- Koodin vastaavuudet muissa tuetuissa koodistoissa ja saman koodiston sisällä, (mediation, translation) on nähty kommenteissa tärkeäksi jatkokehityskohteeksi (v0.9), K4.5, koodien välisiä suhteita on käsitelty mm. luvussa 5.10.
- Tarkimman mahdollisen käsitettävä vastaavan koodin hakemiseen vaiheittain on useita mahdollisuuksia (vaatimus K2.13): voidaan joko määritellä erilaisia suhteita koodiarvojen välille (TQS: LexExplorer: list_associated_target_elements)) tai käyttää jo määrittelyssä olevia hierarkkisten koodistojen käsittelyä tai vaiheittain tarkennettavia hakuavaimia.
- Valintalistan luominen K2.7: TQS:ssä on käytetty kontekstista riippuvaa (esim. tietyn sovelluksen ehdoilla tapahtuvaa) valintalistan luontia. (LexExplorer: get_pick_list, ValueDomainVersion: get_pick_list, ValueDomainVersion: get_pick_list_for_scheme). Vaatimukseen voidaan vastata yksinkertaisesti käyttämällä list_concepts-operaatioita.
- Kentän validoinnissa käytetään is_valid_concept-operaatiota eikä Systemization: validate_concept_expression-operaatiota
- Koodin määrittelemän käsitteen esittäminen sovelluksessa annetuilla ehdoilla K4.6
- Terminologiapalveluiden käyttö yhdistelmäkäsitteiden (composite concept) validointiin, muuttamiseen jne. K2.14.
- Ylläpito-operaatioita ei nähdä tarpeellisiksi tässä määrittelyssä, ja ne on jätetty toteutuskohtaisiksi (v0.9).

4.7 Tekniikkariippumattomiin määrittelyihin tehdyt muutokset

Kuten aiemmin mainittiin, PlugIT-koodistorajapinnan määrittelyksen versiosta 1 (Vaatimukset ja Tekniikkariippumattomat liittymämäärittelyt) tarkennettiin tekniset rajapintamäärittelyt versioon 2 (luku 5). Tässä on lueteltu tärkeimmät versioiden välillä tehdyt muutokset. Muutokset on tehty luvun 4.4 ”operaatioiden ja tasojen yhdistäminen” – periaatteen ja HL7 CTS- (Solbrig 2003), Koodistopalvelu (Koodistopalvelu 2003) -dokumenttien ja PlugIT-hankkeen Ydinrajapintojen yleisten suunnittelukäytäntöjen (Sormunen ym. 2004a) harmonisoinnin pohjalta.

- operaatioiden yhdistäminen parametreja lisäämällä
 - `get_concepts_by_text_lang` yhdistetty `LookupCodesByDesignation:iin`
 - `match_concepts_by_string_lang` yhdistetty `LookupCodes:iin`
 - `list_concepts_lang` yhdistetty `ListCodes:iin`
 - `list_concepts_and_values` yhdistetty `ListCodes:iin` (`propertyCodeList`-parametri, `sortBy`-parametri)
 - `match_concepts_by_substring` yhdistetty `LookupCodes:iin`
 - `match_concepts_by_keywords` yhdistetty `LookupCodes:iin` (voidaan toteuttaa kuin `match_concepts_by_substring`)
 - `match_concepts_by_value` yhdistetty `LookupCodes:iin`
 - `get_concepts_by_text_hier` yhdistetty `LookupCodesByDesignation:iin`
 - `match_concepts_by_substring_hier` yhdistetty `LookupCodes:iin`
 - `match_concepts_by_string_hier` yhdistetty `LookupCodes:iin`
 - `match_concepts_by_keywords_hier` yhdistetty `LookupCodes:iin`
 - `list_concepts_hier` yhdistetty `ListCodes:iin`
 - `list_concepts_and_values_hier` yhdistetty `ListCodes:iin`
 - `list_valid_concepts` yhdistetty `ListCodes:iin`
 - `get_lang_text` yhdistetty `Designation:iin`
- operaatioiden nimien muutokset (HL7 Common Terminology Services, PlugIT-ydinrajapintojen nimeämiskäytäntö):
 - `get_supported_codesets` -> `GetSupportedCodeSystems`
 - `get_concepts_by_text` -> `LookupCodesByDesignation (CTS)`
 - `list_concepts` -> `ListCodes`
 - `is_valid_concept` -> `IsCodeValid (CTS)`
 - `get_concepts_and_values` -> `GetCodes`
 - `get_elements` -> `GetSupportedAttributes`
 - `hier_depth` -> `HierarchyDepth`
 - `get_preferred_text` -> `Designation`
 - `get_values` -> `LookupCompleteCodedConcept`
 - `get_value` -> `LookupProperties`
 - `hier_parent` -> `Parent`
 - `get_status` -> `Status`
 - `is_local_concept` -> `Local`
- nimeäminen: merkkauksen nimet Koodistopalvelu-määrittelyksestä
- `GetSupportedCodeSystems`: kaikki tuetut versiot palautetaan samassa elementissä.
- `LookupCodesByDesignation`: selitteiden avulla tapahtuvat haut yhdistetty yhteen operaatioon, lisätty tarvittavat parametrit (`get_concepts_by_text`, osa `match_concepts_by_stringistä`)
- `LookupCodes`: osittaisella koodiavulla hakuun ja vapaasta sisällöstä tapahtuvaan hakuun muokattu 1.0:n `match_concept_by_stringistä` hakuoperaatio, mm. lisätty `partial`-elementto
- Lisätty tiettyinä aikana voimassa olevien koodien hakua varten `status`-tasoon `current`-parametri.

5 TEKNISET LIITTYMÄMÄÄRITTELYT XML- JA HTTP-RAJAPINNOILLE

Tässä luvussa kuvataan tarkennettu tekninen rajapinta koodistopalveluille, joka perustuu lukujen 3 ja 4 Integroitivaatimuksiin ja Tekniikkariippumattomiin liittyisiin. Luvun sisältö vastaa PlugIT-hankkeessa hyväksyttyä dokumenttia ”Koodistorajapinnat: Tekniset liittymämäärittelyt http- ja XML-rajapinnoille, versio 2”¹. Luvun sisällön pohjalta voidaan toteuttaa avoimia rajapintoja terveydenhuollon sovellusohjelmistoihin. Tarkat rajapintamääritykset edellyttävät lukijalta XML- ja http-tekniikoiden perusteiden tuntemista.

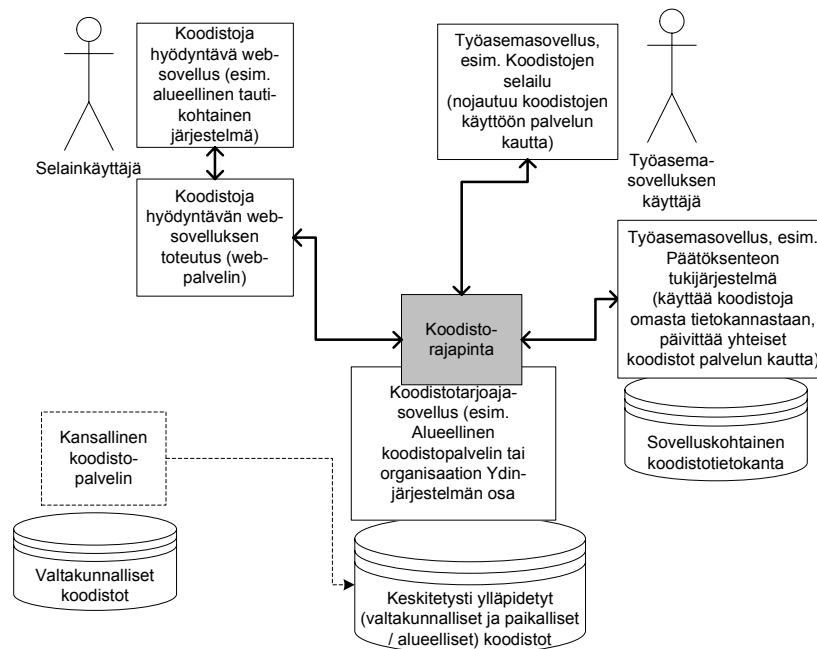
Luvun 4 (version 1.0) rajapintamäärityksiä on harmonisoitu HL7 Common Terminology Services (CTS)-määrityksen (Solbrig 2003) mukaisilla operaatioiden nimillä ja sisällöillä, etenkin CTS:n Vocabulary API-osion osalta (joka vastaa suurelta osin sisällöllisesti PlugIT-projektin 1.0-määritystä) (ks. luvut 2.4 ja 4.7). Tässä määrityksessä on kuitenkin myös säilytetty ne piirteet, joihin versiossa 1.0 vastataan, ja jotka eivät sisälly CTS-määritykseen. Version 1 Tekniikkariippumattomasta määritystä on tarkennettu siten, että liittymätekniikoiksi on valittu http ja XML, kuten useissa muissakin PlugIT-projektin rajapinnoissa (ks. Sormunen 2004a-b). XML-merkkeysten nimeämisessä on hyödynnetty sekä muita PlugIT-projektin ydinrajapintamäärityksiä että Stakesin Koodistopalvelun koodistosiirtosanomien määrityksiä.

Ydinrajapintojen etuja ja perusteita kuvattiin vaatimusten yhteydessä luvussa 1.2. Näitä etuja (pääallekkaisen tietojen syötön ja ylläpidon vähentäminen, ajantasaisten koodistojen käyttö, valmiiksi saatavilla olevien palveluiden ja ohjelmistojen hyödyntäminen, paikallisen sovituksen väheneminen) saavutetaan keskitettyjen koodistopalveluiden osalta toteuttamalla tässä luvussa kuvattuja integroitiratkaisuja sovelluksiin.

PlugIT-projektin koodistomäärittelyissä on hyödynnetty Stakesin koodistopalvelimen tietosisältöön liittyviä määrityksiä (ks. luku 2.5) ja määritelty rajapinnat, joiden avulla sovellukset saavat tarkempia ja sovellusläheisempiä palveluita (esim. ydinjärjestelmältä tai alueelliselta koodistopalvelulta). Näiden rajapintojen toteuttaminen kansallista palvelinta hyödyntäviin ohjelmistoihin helpottaa yhteisesti ja valtakunnallisesti saatavilla olevien koodistojen ”arkikäyttöön” leviämistä sovelluksissa.

Kuva 5.1 havainnollistaa koodistorajapinnan hyödyntämismahdollisuuksia. Lisätietoja ja tarkempaa taustaa, ml. vaatimukset joihin rajapinnat perustuvat, sekä määrityksen tavoitteet ja päärajaukset löytyy tämän raportin aiemmista luvuista.

• ¹ Version 2 mukaiseen määritykseen on tehty korjauksia muutamiin Codeset-rajapinnan operaatiokuvauxiin, joiden operaatiokuvaus-taulukon Interface-kentässä luki virheellisesti ”Codeservice”, korjattu Get-CodesetInfo-operaation käyttötarkoitus, täydennetty virheilmoitusten puuttuvia tietoja ja korjattu kirjoitusvirheitä.



Kuva 5.1. Esimerkkejä koodisto-ydin-rajapinnan hyödyntämisestä.

5.1 Tavoitteet ja päärajaukset

Tämän luvun määritysten tavoitteet noudattavat on täydennetty luvun 3 mukaisista tavoitteista:

1. Liittymämäärittely, jolla koodiston tarjoava sovellus (esim. potilashallinnon ydinjärjestelmä tai alueellinen koodistopalvelin) tarjoaa keskeiset koodistopalvelut niitä tarvitseville sovelluksille.
2. Sovellusten tarvitsemien koodistopalveluiden ohjelmallisten rajapintojen määrittely yhdenmukaisella tavalla.
3. Yhteensopivuus Stakesin koodistopalvelimen sisältöratkaisujen kanssa, kansallisten koodistojen käyttöönotettavuuden edistäminen, ks. luku 2.5.
4. Soveltuvin osin yhteensopivuus HL7 Common Terminology Services-rajapintojen kanssa, ks. luku 2.4.
5. Ratkaisun yleisyys: eri koodistoja (mukaan lukien organisaatio) voidaan hyödyntää saman ratkaisun avulla, ja määritysten mukainen ratkaisu on toteutettavissa eri sovelluksiin ja eri ympäristöihin.
6. Määrittelyt avoimilla tekniikoilla siten, että eri toteutustekniikoilla tehdyt sovellukset voivat hyödyntää määrittelyn mukaisia rajapintoja.
7. Määritysten pilotointi koodistopalveluja tarjoavan, niitä käyttävän ja ratkaisun tilaavan osapuolen kanssa tai referenssitoteutusten tuottaminen (ks. luku 6).
8. Mahdollisten jatkokehityskohteiden tunnistaminen, mm. määrittelyjen kattavuuden lisääminen kohti edistyneempiä terminologiajärjestelmäpalveluita (luku 5.11).

Tekniset määrittelyt noudattava luvun 3.1 päärajauksia:

- keskittyminen pääasiassa organisaation sisällä toimiviin tietojärjestelmiin,
- ratkaisut ovat palvelurajapintoja, ei tuoteta tietokantaintegraatiota eikä dokumentti- tai sanomapohjaista koodistojen siirtoa,
- määriteltävät ohjelmistorajapinnat eivät näy suoraan käyttäjälle käyttöliittymiä,

- ratkaisussa määritellään vain hyödyntämiseen, ei ylläpitoon tai päivitykseen tarvittavia operaatioita.

5.2 Aikaisempien määritysten tarkentaminen

Vaatomukset, joihin tämä määritys vastaa, on kuvattu tarkemmin luvussa 3 (versio 1.0). Keskeisiksi ja keskitettäviksi koodistopalveluiksi nähtiin koodin tunnusteen ja selitteen saanti, koodien erilaiset haut, monikielisten ja hierarkkisten koodistojen käsittely sekä koodien paikallisuus- ja tilatiedon huomiointi.

5.2.1 Tekniikoihin kohdistuvat vaatimukset

Tekniikkariippumattoman määrittelyn version 1.0 rajapinnoista voidaan tuottaa teknisiä määrittelyitä eri integrointitekniikoille (esim. paikallisia kirjastorajapintoja, verkon yli käytettäviä hajautettuja oliopalveluita tai web-sovelluspalveluita). Tämä tekninen määritys vastaa versiossa 1.0 teknisille määrittelyille asetettuihin vaatimuksiin seuraavasti:

- Mikä on tarkka mekanismi minimitason operaatioille koodiston ja koodin käyttöön saamiseksi:
 - koodiston ja koodin yksilöinti nimetyillä parametreilla
- Käytetäänkö XML:ää vai esim. ohjelmointikielten tai liittymämäärittelyjen tietorakenteita, erotetaan tietosisältö XML-merkkauksella toiminnoista (aina tai esim. vain freeElements-tasolla),
 - Liittymä, operaatio ja kaikki parametrit ovat http-kutsussa välitettävää XML-merkattua tietoa. Vastaukset ovat http-vastauksessa palautuvia yksinkertaisia XML-merkattuja dokumentteja.
- Kuinka määritellyt toiminnot (operaatiot) ja niiden tarvitsema ja palauttama tietosisältö ja tietotyypit teknisesti määritellään (tietotyypeissä voidaan perustellusti poiketa tässä kuvatuista),
 - XML-elementeissä ja -attribuuteissa olevan tiedon muoto on selostettu kunkin operaation kuvauksessa. XML-dokumenteissa ja DTD-esimerkeissä tietoa käsitellään merkkijonoina ja esim. elementin numeerisuus käy ilmi kuvauksesta ja esimerkeistä (ilman tietorytymäärittystä). Eri operaatioista tai kaikista XML-määrittelyistä voidaan muodostaa XML Schema tai DTD-määrittelyt; operaatiokohtaisia DTD-määrittelyjä on käytetty operaatiokuvausten jälkeen esimerkkeinä.
- Käytetäänkö tietosisällön merkkauksessa samaa XML-muotoa tai sen osajoukkoa, jota käytetään Stakesin koodistopalvelu-määrittelyissä (Koodistopalvelin 2003: SOAP Body-osion sisällä oleva XML-määrittely),
 - Käytetään (Koodistopalvelu 2003)-mukaisia elementtien ja attribuuttien nimiä ja malleja soveltuvissa tietosisällöissä.
- Kuinka koodistohyödyntäjä voi paikallistaa ja saada käyttöönsä koodistotarjoajan tarjoaman palvelun,
 - Kutsujen http-osoite on parametroitava paikallisesti, myös nimipalvelun tai palvelurekisterin käyttö olisi mahdollista.
- Miten vapaaehtoisiksi merkityt parametrit käsitellään, esim. lähetetäänkö parametria lainkaan vai onko niiden tilalla tyhjä arvo tai puuttuvaa tarkoittava arvo, ja kuinka tyhjät arvot teknisesti toteutetaan.
 - Vapaaehtoiset parametrit voivat puuttua kutsuista kokonaan. Puuttuva parametri vastaa ”tyhjää” tai oletusarvoa. Operaatiokohtaisesti on määritelty, palautuuko ”tyhjiin” parametrien tai paluuarvojen kohdalla esim. tyhjä vastaus tai virheilmoitus. Palvelun toteutusten on kuitenkin ymmärrettävä ”pakollisiksi” merkityt elementit (ks. luku 5.5).

5.2.2 Rajapintojen ja tasojen yhdistäminen

Version 1 koodipalvelu-, koodisto- ja koodirajapinnat on säilytetty tässä määrittelyssä. Sen sijaan operaatioiden määrää on vähennetty lisäämällä eri tasojen vaatimia parametreja operaatioihin, kuten luvussa 4.4 on kuvattu. XML:n käyttö mahdollistaa tällaisen joustavan ratkaisun ja mm. ”tarpeettomien” parametrien jättämisen pois kutsuista. Lista tärkeimmistä muutoksista on luvussa 4.7.

5.3 Valitut tekniikat ja niiden käyttötavat

Tässä dokumentissa avoimiksi liittymätekniikoiksi on valittu http ja XML, jotka on havaittu käyttökelpoisiksi ja laajalti hyödynnettäviksi myös muissa PlugIT-projektin integrointikohteissa. Nämä tekniikat vastaavat esitettyihin avoimuusvaatimuksiin ja niillä on suhteellisen kevyt toteuttaa sovitimia olemassa oleviin sovelluksiin. Useissa saaduissa kommentteissa tämä tekniikka on nähty ensimmäisenä määrittelytarpeena.

Koodistoja tarjoavan palvelun rajapinnat on määritelty tilattomina palveluina, mikä tarkoittaa sitä että kaikki tarvittava tieto (esim. tarvittaessa koodiston ja koodin tunniste) liikkuu parametrimina jokaisessa ohjelmistokutsussa koodistotarjoajalle. Paluuarvossa ei yleensä toisteta kaikkia pyynnöissä esitettyjä parametreja.

Palveluun tehtävien kutsujen yleinen muoto on http-kutsu, jonka POST-sisältönä on XML-dokumentti, joka sisältää tarvittavat parametrit. Kutsun vastaus palautuu vastauksena tulevassa http-viestissä. Sekä kutsuissa että vastauksissa siirretään text/xml-muotoista (content-type) tietoa, ja sekä koodistotarjoajan että koodistohyödyntäjän on tuettava UTF-8-merkistökoodausta viesteissä (muuta koodauksia voidaan tukea tuotekohtaisesti ja dokumentoituna).

Pyyntöjen ja vastausten XML-elementtien järjestys on tämän dokumentin operaatiokuvausten ja DTD-esimerkkien mukainen. Yksittäisten elementtien eri attribuuttien järjestys (jos elementillä on useita attribuutteja) on vapaa.

Koodistohyödyntäjän toteutusten on toimittava siten, että vaikka koodistotarjoaja palauttaisi tai lähettäisi sellaisia elementtejä tai attribuutteja, jotka eivät sisälly esim. perustason määrittelyyn, jätetään ”ei-määritelty” elementit ja attribuutit huomiotta. Tämä pätee erityisesti vapaa tietosisältötasolla, jossa palautettavissa termItemEntry-elementeissä voidaan käyttää (Koodistopalvelu 2003)-koodistosiirtomäärittelysten mukaista sisältöä.

HL7:n CTS-määrittely (luku 2.4) sisältää myös esimerkin Javan kautta tuotetun WSDL-rajapinnan määrittelystä, joka hyödyntää samoja pohjatekniikoita kuin tämä määrittely, mutta saaduissa osapuolten kommentteissa XML ja http nähtiin nopeimmin hyödynnettävinä avoimina tekniikoina.

Määrittely ei ota kantaa salaukseen tai palveluiden käyttöoikeuksiin. Koodistotieto ei yleensä ole suojattavaa, mutta palvelun käyttöä on mahdollista rajoittaa mm. vain tiettyihin sovelluksiin lisäparametrien ja –operaatioiden avulla (esim. Sormunen ym. 2004b).

5.4 Tekniset sovellusosat, niiden vastuut ja tiedonvälitys

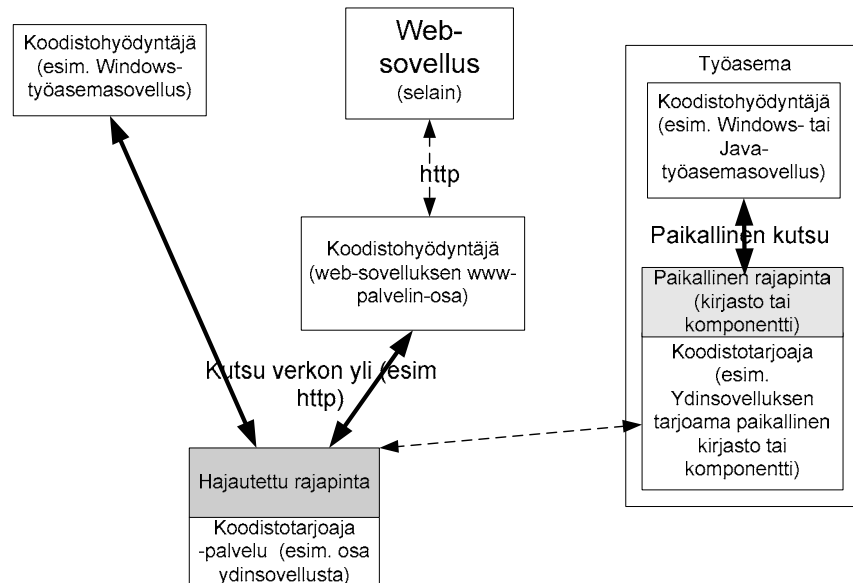
Määriteltävässä ratkaisussa osapuolina ovat:

1. *Koodistotarjoaja*, sovellus tai komponentti, joka tarjoaa määriteltyt rajapinnat tai rajapintajoukot ja vastaa ratkaisussa sen vastuulle määritellyistä seikoista. Koodistotarjoaja on teknisesti toteutettu http-viestejä kuuntelevana palvelimena, joka ymmärtää ja palauttaa XML-muotoisia kutsuja ja paluuarvoja. Koodistotarjoaja toimii koodistojen ja kooditietojen keskitettynä ylläpitopaikkana.
2. *Koodistohyödyntäjä*, sovellus tai komponentti, joka käyttää koodistotarjoajan tarjoamia palveluita. Koodistohyödyntäjä ottaa yhteyttä koodistotarjoajaan http-kutsuilla, joiden sisältönä ole-

viin XML-elementteihin se asettaa tarvittavat parametrit. Se myös purkaa ja käsittelee koodistotarjoajan palauttavat paluuarvot.

Koodistohyödyntäjän ja koodistotarjoajan väliset kutsut kulkevat verkon yli kuvan 5.2 ”hajautettu rajapinta”-osan mukaisesti. Rajapinnan pohjalta voidaan rakentaa myös paikallisia (esim. DLL-kirjasto) rajapintoja tai sovittimia, joilla http- ja mahdollisesti myös XML-yksityiskohdat piilotetaan paikallisen rajapinnan hyödyntäjältä.

Ratkaisun periaatteena on koodistohyödyntäjän kannalta synkroninen pyyntö/vastaus, jossa koodistohyödyntäjä saa lähettämäänsä kyselyyn välittömästi vastauksen (tai virheen) koodistotarjoajalta.



Kuva 5.2. Esimerkkejä määritellyn palvelun teknisistä toteutuksista ja käyttötavoista, luvun 5 määrittysten mukainen osio on kuvan osan ”hajautettu rajapinta” käyttö.

5.5 Operaatio- ja parametrimääritykset

Koodistotarjoajan palvelut ryhmitellään seuraaviin rajapintoihin, kuten luvussa 4:

- koodistopalvelun toteutukseen liittyvät palvelut: *Codeservice*-rajapinta - liittyvät itse palvelun toteutukseen ja hyödyntämiseen.
- koodistoon liittyvät palvelut: *Codeset*-rajapinta -kohdistuvat yksilöityyn (yhteen) koodistotarjoajan tarjoamaan koodistoon. Tarvittaessa teknisessä toteutuksessa välitetään koodiston tunniste (in-) parametrina eri operaatioille.
- koodiin liittyvät palvelut: *Code*-rajapinta - liittyvät tunnistettuun ja yksilöityyn koodiin.

Kaikkien rajapintojen ja operaatioiden tarvitsemat ja palauttavat tiedot on määritelty yksinkertaisina XML-rakenteina. Käytettyjen XML-elementtien kuvaukset löytyvät operaatiokuvausten lisäksi myös liitteestä 4.

Kustakin operaatiosta esitetään kutsut seuraavasti:

Interface: request-elementin interface-elementin sisältö

Method: request-elementin method-elementin sisältö

Param: request-elementin param-elementin sisällön kuvaus.

Palvelukutsujen yleinen muoto ja elementtien järjestys on siis (DTD-notaatiolla):

```
<!ELEMENT request (interface,method,param?)>
```

ja kaikki vastaukset (ml. virheilmoitukset) palautuvat response-elementissä.

Paluuarvot esitetään response-elementin sisällön kuvauksina.

Kussakin operaatiokuvauksessa on lisäksi käyttötarkoitus, sanallinen kuvaus, pakollisuus (mitä parametreja ja millaista toiminnallisuutta operaation toteutusten on eri palvelutasoilla toteutettava), mahdolliset lisätiedot, yleisimmät poikkeukset sekä viittaus vaatimuksiin, joihin operaatio liittyy. Vaatimusnumerot viittaavat vaatimusedokumenttiin eli versioon 1 (luku 3). Useista operaatioista on myös parametrien ja paluuarvojen DTD-kuvaus (huom. kuvaukset eri operaatioissa ovat erilaisia myös samoille elementeille, koottu DTD voidaan muodostaa yhdistämällä eri operaatioiden kuvaukset). DTD-notaation osaaminen ei kuitenkaan ole edellytyksenä dokumentin lukemiselle ja XML:n muodostamiselle. Operaatiomäärittysten jälkeen on usein myös esimerkkejä operaation kutsuista ja paluuarvoista eri palvelutasoilla. Operaatiokuvauksissa elementeistä on vahvennettu operaation toteutuksessa vähintään vaadittavat. Virhetilanteita käsitellään luvussa 5.9.

Elementtien kohdalla maininta ”vapaaehtoinen” tarkoittaa, että se voidaan jättää haluttaessa pois tehtävistä kutsuista, mutta koodistotarjoajan toteutuksen on toteutettava se siten kuin kohdassa ”pakollisuus” on mainittu.

Koodiston id-arvona sen yksilöinnissä voidaan käyttää tarkkaa (myös version sisältävää) OID-koodia tai koodiston yleistä OID-koodia.

5.5.1 Määrittelyn palvelutasot ja niillä vaaditut piirteet

Koodistotarjoaja ja –hyödyntäjä voivat toteuttaa joukon erilaisia palvelutasoja tämän määrittelyn pohjalta. Yleensä vain jotkin tasoista toteutetaan, ja jo minimitasolla saavutetaan keskitettyjen koodistojen etuja. Tasoja voidaan rakentaa sovelluksiin esim. yksi kerrallaan, ja myös uusia tasoja voidaan lisätä määrittelyyn myöhemmissä versioissa.

- *minimitaso* – vain keskitetyn koodistopalvelun välttämättömät operaatiot olemassa olevien sovellusten helpon liitettävyyden kannalta (vaatimus K1.5): koodiston ja koodin käyttöön saanti, koodien listaus, minimitietosisältö (koodi ja selite), selitteen saanti koodilla, koodien saanti selitteellä
- perustaso ”base” – minimitason lisäksi osamerkkijonolla haku selitteestä ja koodista, tuettujen palveluiden ja koodistojen selville saanti (vaatimus K5.1), palvelun ja koodistojen (vapaaehtoiset) hallinnolliset tiedot (K3.2, K2.4),
- monikieliset koodistot ”*multilingual*”, useiden tai määriteltyjen kielten tuki selitteissä ja muussa tietosisällössä, (vaatimus K5.2), sisältää kieltä tarkentavan attribuutin lang (ks. liite 3). Vapaan tietosisällön yhteydessä kieli voi liittyä ainakin elementteihin ”abbreviation”, ”shortname” ja ”longname”.
- vapaa tietosisältö ”*freeElements*” - perustietojen (koodiarvo, selite) lisäksi minkä tahansa koodien sisältämien kenttien käyttö eri operaatioissa (vaatimus K5.5),
- edistyneet haut ”*advSearch*” - haku koodiston koko sisällöstä ml. alimerkkijonot, haku avainsanoilla, hakuehtojen kohdistaminen haluttuihin kenttiin, useiden hakuehtojen käyttö (vaatimus K5.4),

- koodistohierarkian käsittely ”*hierarchy*” (vaatimus K5.3),
- paikallisten koodien ja tilan huomiointi ”*status*” (vaatimus K2.15), sisältää kolme hakuja ja listauksia rajoittavaa elementtiä: status, local ja current. Oletusarvoisesti palvelu palauttaa myös paikalliset ja poistetuiksi merkityt koodit. Viittaukset myös tällaisiin koodeihin sallitaan, mutta varsinkin alueellisesti tai eri koodipalveluja käyttävien sovellusten tiedonsiirrossa tulisi välttää etenkin paikallisten koodien käyttöä. Status-tason toteuttaminen on erittäin suositeltavaa käyttötilanteissa, joissa käyttäjälle tulisi esim. valintaa varten näyttää vain voimassaolevat koodit. Status-arvolla ei ole suoraa linkkiä ”vapaa tietosisältö” tason mahdollisiin koodien voimassaolopäivämääriin (beginDate, expirationDate) elementteihin, vaan Status-tieto käsitellään niistä erillisenä. Voimassaolopäivämäärien suhteen tapahtuvien hakujen toteuttamiseksi määrittelyyn on versiossa 2 lisätty current-parametri.

Eri tasojen tarkat vaatimukset toteutuksille on kuvattu operaatiokuvauksissa ja kootusti liitteessä 3.

Hakujen kutsuparametreissa esim. teksti ”palvelun on tunnistettava status, local-ja parentId-elementit ja palautettava niitä vastaavat tulokset” tarkoittaa, että hakua rajataan näillä elementeillä, mutta palautettava tietosisältö on normaali, eikä sisällä status ja local tietoja, ellei näin ole erikseen (muilla parametreilla) määritelty..

5.6 Codeservice-rajapinta

5.6.1 GetSupportedCodeSystems - koodistotarjoajan tukemat koodistot

Taulukko 5.1. CodeService-rajapinnan operaatiot.

Method	GetSupportedCodeSystems
Interface	Codeservice
Käyttötarkoitus	Koodistotarjoajan tukemien koodistojen listaus
Param	ei param-elementtiä
Response	Joukko termSystem -elementtejä, joista kunkin • id-attribuutissa koodiston tunniste, (pakollinen) • language-attribuutissa kielikoodi (jos koodistossa tuettuna vain yksi kieli, tai oletuskieli) ”multilingual” • version-attribuutissa tarvittaessa paikallinen versio tai tarkempi tunniste
Kuvaus	Palvelu palauttaa tiedot kaikista koodistoista ja niiden versioista, joita se tukee. Kaikki koodistojen versiot (kokonaiset OID-polut, mukaan lukien versionumero) palautetaan.
Pakollisuus	Pakollinen perustasolla ”base”. Monikieliset koodisto ”multilingual”-tasolla palautettava termSystemin language-attribuutti, jos saatavilla ja sovellettavissa.
Lisätietoja	Jos palvelu tukee useita versioita samasta koodistosta ja versioilla on omat OID-koodit, kustakin versiosta palautetaan yksi termSystem-elementti. Tällöin koodiston yleinen OID palautetaan yhdessä elementissä jonka version-attribuutissa voi olla oletusversion OID. Asiakas voi viitata koodistoon pelkällä koodisto-OID:llä (ilman versiota), jolloin palvelu palauttaa oletusversion (yleensä uusin versio) mukaisia vastauksia. Paikallisten koodistojen (ei-OID-yksilöityjen) käyttö voidaan sallia toteutuksissa. Tällöin niillä on oltava muu yksikäsitteinen tunniste, jonka koodistohyödyntäjä tuntee. Jos koodistossa ja palvelussa on tuettu useita kieliä, voidaan palauttaa oletuskieli. Muiden kielten selville saantiin voidaan multilingual-tasolla käyttää ListLanguages-operaatiota.
Poikkeukset	GeneralFailure, NotImplemented
Vaatimukset	K3.1, K2.2-OID

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>CodeService</interface>
  <method>GetSupportedCodeSystems</method>
</request>
```

Vastaus DTD

```
<!ELEMENT response (termSystem+)>
<!ATTLIST termSystem
  id CDATA #REQUIRED
  language CDATA
  version CDATA>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <termSystem id="1.2.246.537.6.1.1996">ICD-10 1996</termSystem>
  <termSystem id="1.2.246.777.5.164.2003.1" version="1.0">
    Lääkkeenantolaite 2003</termSystem>
</response>
```

5.6.2 GetSupportedServices - koodistotarjoajan palvelut

Method	GetSupportedServices
Interface	Codeservice
Käyttötarkoitus	Koodistotarjoajan tukemien palvelutasojen listaus
Param	ei param-elementtiä
Response	Joukko service-elementtejä, joista kunkin id-attribuutissa palvelutason tunniste, (pakollinen), jokin seuraavista "base" – perustaso "multilingual" – monikieliset koodistot "freeElements" – vapaa tietosisältö "advSearch" – edistyneet haut "hierarchy" – koodistohierarkian käsittely, huom. hierarkkisia koodistoja voidaan käsitellä myös "ei-hierarkkisinä" "status" – paikallisten koodien ja tilan huomiointi - tekstisisältönä palvelutason näytettävä nimi (vapaaehtoinen, suositellaan) - version-attribuutissa viittaus standardiin, jossa määritellystä palvelutasosta on kyse, tämän dokumentin osalla "2.0", voidaan käyttää myös määrittelyn OID-koodia.
Kuvaus	Palvelu palauttaa tiedot siitä, mitä koodistopalveluita se tukee. Palvelun on täytettävä palvelutasolle tässä dokumentissa asetetut vaatimukset, jotta se voi palauttaa kyseisen palvelutason tunnisteiden.
Pakollisuus	Pakollinen perustasolla "base".
Lisätietoja	Minimitasossa tätä operaatiota ei vaadita, palvelu voi palauttaa virheen tai tyhjän joukon get_supported_services-operaatiosta. Versionumerosta ei voi päätellä, että palvelu tukee edellisiä versioita, vaan jos tuetaan useita erilaisia määrittäversioita ja versiot halutaan eritellä, kutakin tuettua versiota kohti palvelu palauttaa service-elementin.
Poikkeukset	GeneralFailure, NotImplemented
Vaatimukset	K2.8

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>CodeService</interface>
  <method>GetSupportedServices</method>
</request>
```

Vastaus DTD

```
<!ELEMENT response (service*)>
<!ATTLIST service
  id CDATA #REQUIRED
  version CDATA>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <service id="base" version="2.0">Perustaso</service>
  <service id="multilingual" version="2.0"></service>
  <service id="freeElements" version="2.0"></service>
  <service id="advSearch">Edistyneet haut</service>
  <service id="hierarchy">Koodistohierarkia</service>
  <service id="status" version="2.0"></service>
</response>
```

5.6.3 GetInfo - koodistotarjoajan hallinnolliset ja yhteystiedot

Method	GetInfo
Interface	Codeservice
Käyttötarkoitus	Koodistotarjoajan (palvelimen) tiedot ja kuvaus
Param	ei param-elementtiä
Response	Voi palauttaa tyhjän response-elementin. server-elementti (vapaaehtoinen) jonka sisältönä tuotteen nimi - version-attribuutissa (vapaaehtoinen) tuotteen versio - description-elementti (vapaaehtoinen) - Joukko services-elementtejä (vapaaehtoinen, toistuva) - Joukko termSystem-elementtejä (vapaaehtoinen, toistuva)
Kuvaus	Description-elementin sisällössä voi olla mm. tuote ja versio, toteuttaja sekä koodistotarjoajan toteuttajan yhteystiedot. Operaatio voi myös palauttaa joukon service- ja/tai termSystem-elementtejä, jolloin GetInfon paluuarvo sisältää myös GetSupportedServices ja GetSupportedCodeSystems-operaatioiden mukaiset paluuarvot.
Pakollisuus	Pakollinen perustasolla "base".
Lisätietoja	
Poikkeukset	GeneralFailure, NotImplemented
Vaatimukset	K3.2

Vastaus DTD

```
<!ELEMENT response (server?,description?,service*,termSystem*)>
```

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>CodeService</interface>
  <method>GetInfo</method>
</request>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <server version="1.0">My Code Provider</server>
  <description>University of Kuopio Terminology API according to PlugIT
    specifications. 2004.
  </description>
  <service id="base" version="2.0">Perustaso</service>
  <termSystem id="1.2.246.537.6.1.1996">ICD-10 1996</termSystem>
</response>
```

5.7 Codeset-rajapinta

Koodisto yksilöidään Codeset-rajapinnassa param-elementin sisällä termSystem-elementin id-attribuutin avulla.

5.7.1 LookupCodesByDesignation - koodien haut selitteillä

Taulukko 5.2. Codeset-rajapinnan operaatiot.

Method	LookupCodesByDesignation
Interface	Codeset
Käyttötarkoitus	Koodien hakeminen selitteen avulla
Param	termSystem-elementti, jonka ▪ id-parametrissa koodiston tunniste find-elementti, jonka sisällä ▪ matchText-elementin tietosisältönä hakuarvo (ilman esim. jokerimerkkejä), matchTextin attribuutit ○ language, hakuarvon kielen tunniste (multilingual) ○ partial: 0-(oletus) haetaan vain tarkkoja osumia, 1-haetaan alusta hakuarvoon sopivia osumia (base), 2-haetaan missä tahansa elementin osassa (myös keskellä) olevia osumia) (advSearch) ▪ status-elementti, jos haetaan koodin tilan perusteella (status), sisältönä 1 (=active), 2 (=delete) tai 0 (=work), oletuksena ei huomioida ▪ local-elementti, jos haetaan koodin paikallisuuden perusteella (status), sisältönä 0 (=ei-paikallinen) tai 1 (=paikallinen), oletuksena ei huomioida ▪ current-elementti, jos haetaan voimassa olevia koodeja (status), sisältönä päivämäärä, jolloin voimassa olevia haetaan ▪ parentId-elementti, jos haetaan alihierarkiasta (hierarchy) sortBy-elementti (perustaso), paluuarvojen lajittelukenttä, sisältö: ks. liite 4 display-elementti (vapaaehtoinen, freeElements), palautettavat elementit, sisältää ▪ propertyCodeList-elementin, jossa yksi tai useampia ○ property-elementtejä, sisältönä palautettavan elementin nimi, attribuuttina tarvittaessa (multilingual) ▪ language-attribuutti, arvon kielen tunniste

Response	Lista joko term- tai termItemEntry-elementtejä. termItemEntry-elementtejä, jos toteutetaan freeElements-taso ja hakuparametreissa on määritelty display-parametri. term-elementin sisältönä (pakollinen) koodin näytettävä selite (shortname), attribuuttina ▪ id – koodin tunniste (pakollinen) ▪ language – selitteen kieli (multilingual-tasolla) termItemEntry – elementin ▪ attribuuttina id – koodin tunniste (pakollinen) ▪ sisältönä joukko ○ attribute-elementtejä (ks. liite X), joista palautettava ainakin ▪ type-attribuutti (ominaisuuden nimi) ▪ tietosisältö (näytettävä arvo elementistä) ▪ language-attribuutti, jos halutaan tietyn kielinen arvo ja tuetaan eri kieliä
Kuvaus	Palvelu palauttaa ne koodit, joiden selite vastaa annettua hakuarvoa ja jotka vastaavat muita hakuehtoja. Pienet ja isot kirjaimet vastaavat toisiaan hakuarvoissa.
Pakollisuus	Minimitasolla pakollisia termSystem ja sen id, find-elementti, ja matchtext-elementti. Palvelun palautettava tarkasti matchTextiä vastaavien (partial=0 oletus, ei tarvitse antaa) koodien tunnisteet ja selitteet oletuskielillä. Perustasolla ”base” palvelun tunnistettava matchTextin partial=0 ja partial=1-arvot ja palautettava tulokset niiden mukaisesti. Palvelun tunnistettava sortBy-elementti ja sen sisältönä id tai shortname, ja palautettava arvot halutun ominaisuuden mukaan lajiteltuna. Monikieliset koodistot ”multilingual”-tasolla palvelun tunnistettava matchTextin language-attribuutti. Jos lisäksi toteutettu vapaa tietosisältö, tunnistettava property-elementin attribuuttina kieli jos saatavilla. Palautettava term ja attribute-elementtien language-attribuutissa kyseisen elementin kieli, jos saatavilla. Hierarkkisten koodistojen käsittely ”hierarchy” tasolla palvelun tunnistettava parentId-elementti, palautettava juuren alla olevalla tasolla löytyneet tulokset. Paikalliset koodistot ja tila ”status” tasolla palvelun tunnistettava status, current ja local elementit ja palautettava niitä vastaavat tulokset jos sovellettavissa. Edistyneet haut ”advSearch”-tasolla palvelun tunnistettava partial=2-arvo ja palautettava hakuarvon osumat sen mukaisesti. Palvelun on myös hyväksyttävä useita find-elementtejä. Vapaa tietosisältö ”freeElements” tasolla palvelun tunnistettava display elementti ja palautettava termItemEntry-elementtejä term-elementtien sijaan. sortBy-arvona voi tällä tasolla olla mikä tahansa palvelun tukema ominaisuus, jos tuetaan sekä freeElements että advSearch-tasoa.
Lisätietoja	Language-attribuutti on SFS-ISO-639 mukainen kielen koodi. Jos kutsussa ei ole display-elementtiä, palautuu term-elementtejä, muussa tapauksessa termItemEntry-elementtejä. Jos ”advSearch”-tasolla käytetään useita find-elementtejä, palautetaan vain tulokset, jotka täyttävät kaikkien find-elementtien hakuehdot.
Poikkeukset	Paluuarvo on tyhjä response-elementti, jos hakuarvoja ei vastaa mikään koodi koodistossa. GeneralFailure, NotImplemented (myös jos annetaan partial-arvo jota ei tueta), MissingParameter, UnknownLanguage, TooManyCodes (jos hakuarvo palauttaisi liikaa tuloksia (toteutuskohtainen)), UnknownAttribute, UnknownCodeSystem
Vaatimukset	K2.5

Kutsu DTD

```
<!ELEMENT request (interface,method,param)>
<!ELEMENT param (termSystem,find+,sortBy?,display?)>
<!ELEMENT find (matchtext,status?,local?,current?,parentId?)>
<!ATTLIST matchtext
  language CDATA
  partial CDATA>
<!ELEMENT display (propertyCodeList)>
<!ELEMENT propertyCodeList (property+)>
<!ATTLIST property
  language CDATA>
```

Esim 1 kutsu (minimitaso)

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>LookupCodesByDesignation</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"></termSystem>
    <find>
      <matchText>lavantauti</matchText>
    </find>
  </param>
</request>
```

Esim 2 kutsu (edistyneet haut, vapaa tietosisältö, monikieliset, paikallisuus ja tila, hierarkia)

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>LookupCodeByDesignation</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <find>
      <matchText language="fi" partial="2">avantaut</matchText>
      <status>1</status>
      <local>0</local>
      <current>2004-04-16</current>
      <parentId>A00-A09</parentId>
    </find>
    <sortBy>id</sortBy>
    <display>
      <propertyCodeList>
        <property>shortname</property>
      </propertyCodeList>
    </display>
  </param>
</request>
```

Vastaus DTD

```
<!ELEMENT response (term*|termItemEntry*)>
<!ATTLIST term
  id CDATA #REQUIRED
  language CDATA>
<!ELEMENT termItemEntry (attribute*)>
<!ATTLIST termItemEntry
  id CDATA>
<!ATTLIST attribute
  type CDATA
  dataType CDATA
  language CDATA>
```

```

createDate CDATA
beginDate CDATA
expirationDate CDATA
lastModifiedDate CDATA
lastModifiedBy CDATA>

```

Esim. vastaus 1 (minimitaso)

```

<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <term id="A01.0">Lavantauti</term>
</response>

```

Esim. vastaus 2 (edistyneet haut, vapaa tietosisältö, monikieliset)

```

<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <termItemEntry id="A01.0">
    <attribute type="shortname" language="fi">Lavantauti</attribute>
  </termItemEntry>
  <termItemEntry id="A01.1" >
    <attribute type="shortname" language="fi">Pikkulavantauti
  </attribute>
  </termItemEntry>
  ...
</response>

```

5.7.2 ListCodes - koodien listaus

Method	ListCodes
Interface	Codeset
Käyttötarkoitus	Koodien listaus
Param	termSystem-elementti, jonka ▪ id-parametrissa koodiston tunniste ▪ language-parametrissa kielen tunniste, jos halutaan tietyn kielen selite (ja toteutetaan multilingual-taso) howMany-elementti: kerralla palautettavien maksimimäärä (vapaaehtoinen, jolloin palvelin palauttaa kaikki tai toteutuskohdaisen oletusmäärän koodeja). from-elementti: ensimmäisen palautettavan koodin koodiarvo (tunniste), puuttuu jos haetaan alusta lähtien status-elementti, jos listausta rajataan koodin tilan perusteella (status), sisältönä 1 (=active), 2 (=delete) tai 0 (=work) local-elementti, jos haetaan koodin paikallisuuden perusteella (status), sisältönä 0 (=ei-paikallinen) tai 1 (=paikallinen) current-elementti, jos haetaan voimassa olevia koodeja (status), sisältönä päivämäärä, jolloin voimassa olevia haetaan parentId-elementti, jos haetaan alihierarkiasta (hierarchy) sortBy-elementti (perustaso), paluuarvojen lajittelukenttä, sisältö: ks. liite 3 display-elementti (vapaaehtoinen, freeElements), palautettavat elementit, sisältää ▪ propertyCodeList-elementin, jossa yksi tai useampia ○ property-elementtejä, sisältönä palautettavan elementin nimi, attribuuttina tarvittaessa (multilingual) ▪ language-attribuutti, arvon kielen tunniste (jos toteutetaan multilingual-taso freeElementsin lisäksi)

Response	Lista joko term- tai termItemEntry-elementtejä. termItemEntry-elementtejä käytetään vain jos toteutetaan freeElements-taso ja hakuparametreissa on määritelty display-parametri. term-elementin sisältönä (pakollinen) koodin näytettävä selite (shortname), attribuuttina ▪ id – koodin tunniste (pakollinen) ▪ language – selitteen kieli (multilingual-tasolla) termItemEntry – elementin ▪ attribuuttina id – koodin tunniste (pakollinen) ▪ sisältönä joukko ▪ attribute-elementtejä (ks. liite 3), joista palautettava ainakin ○ type-attribuutti (ominaisuuden nimi) ○ tietosisältö (näytettävä arvo elementistä) ○ language-attribuutti, jos tuetaan useita kieliä ja palautetaan halutun kielinen arvo from-elementin arvona on seuraavan vielä palauttamattoman koodin tunniste, jota voidaan käyttää from-parametrina jatkohaussa. Jos kaikki tulokset on palautettu, from-elementti puuttuu vastauksesta.
Kuvaus	Palvelu palauttaa listan koodiston koodeista koodiarvon (oletus) tai sortBy-parametrin määrittelemässä järjestyksessä. Listaukseen mukaan otettavia koodeja voi rajata lisäparametrien avulla. Peräkkäisillä hauilla voidaan jatkaa listausta ensimmäisestä vielä palauttamattomasta koodista. Jos how_many-parametri puuttuu, koodistotarjoaja palauttaa toteutuskohdaisen oletusmäärän vastauksia.
Pakollisuus	Minimitasolla pakollisia termSystem ja sen id, from-elementti, howMany-elementti. Palautetaan term-elementtejä, id ja sisältö (oletuskielillä) pakollisia. Palvelun palautettava from-elementti, jos kaikkia tuloksia ei ole palautettu. Perustasolla ”base” palvelun tunnistettava sortBy-elementti ja sen sisältönä id tai shortname, ja palautettava arvot sortBy-ominaisuuden mukaan lajiteltuna. Monikieliset koodistot ”multilingual”-tasolla palvelun tunnistettava termSystemin language-attribuutti. Jos lisäksi toteutettu vapaa tietosisältö, tunnistettava property-elementin attribuuttina kieli jos saatavilla. Palautettava term ja attribute-elementtien language-attribuutissa kyseisen ominaisuuden kieli ja selite vastaavalla kielellä, jos saatavilla. Hierarkkisten koodistojen käsittely ”hierarchy” tasolla palvelun tunnistettava parentId-elementti, listattava vain juuren alla olevalla tasolla löytyneitä tuloksia. Paikalliset koodistot ja tila ”status” tasolla palvelun tunnistettava status, current ja local elementit ja palautettava niitä vastaavat tulokset, jos sovellettavissa Edistyneet haut ”advSearch”: ks. Vapaa tietosisältö Vapaa tietosisältö ”freeElements” tasolla palvelun tunnistettava display elementti ja palautettava termItemEntry-elementtejä term-elementtien sijaan, lisäksi ks. ”multilingual”, sortBy-arvona voi tällä tasolla olla mikä tahansa palvelun tukema ominaisuus, jos tuetaan sekä freeElements että advSearch-tasoja.
Lisätietoja	Koodistohyödyntäjä ei saa muuttaa sortBy- tai parentId- parametria haun ja jatko haun välissä.
Poikkeukset	TooManyCodes (jos how_many-parametri ylittää koodistotarjoajan toteutuskohdaisen maksimimäärän), NotImplemented, MissingParameter, TooManyCodes (jos howMany on liian suuri (toteutuskohdainen)), UnknownAttribute, UnknownCodeSystem, UnknownLanguage, GeneralFailure
Vaatimukset	K2.3, K2.7

Kutsu DTD

```
<!ELEMENT request (interface,method,param)>
<!ELEMENT param (termSystem,howMany?,from?,status?,local?,current?,
  parentId?,sortBy?,display?)>
<!ELEMENT display (propertyCodeList)>
<!ELEMENT propertyCodeList (property+)>
<!ATTLIST property
  language CDATA>
```

Kutsu esim. (minimitaso)

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>ListCodes</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <howMany>10</howMany>
    <from>A92.2</from>
  </param>
</request>
```

Vastaus DTD

```
<!ELEMENT response ((term*|termItemEntry*),from?)>
<!ATTLIST term
  id CDATA #REQUIRED
  language CDATA>
<!ELEMENT termItemEntry (attribute*)>
```

Vastaus esim. (minimitaso)

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <term id="A92.2">Venezuelan hevosaivokuume</term>
  <term id="A92.3">Länsi-Niilin kuume</term>
  ...
  <from>A94</from>
</response>
```

5.7.3 LookupCodes - koodien muut haut

Method	LookupCodes
Interface	Codeset
Käyttötarkoitus	Koodien haku osittaisella koodiarvolla, vapaasta tietosisällöstä, useista elementeistä tai useilla hakuarvoilla
Param	termSystem -elementti, jonka ▪ id-parametrissa koodiston tunniste find -elementti ("advSearch" tasolla myös useita), jonka sisällä ▪ matchText-elementin tietosisältönä hakuarvo (ilman esim. jokerimerkkejä), matchTextin attribuutit ▪ language, hakuarvon kielen tunniste (multilingual), vapaaehtoinen ▪ partial: 0-(oletus) haetaan vain tarkkoja osumia, 1-haetaan alusta hakuarvoon sopivia osumia, 2-haetaan missä tahansa elementin osassa (myös keskellä) olevia osumia) (advSearch) ▪ status-elementti, jos haetaan koodin tilan perusteella (status), sisältönä 1 (=active), 2 (=delete) tai 0 (=work) ▪ local-elementti, jos haetaan koodin paikallisuuden perusteella (status), sisältönä 0 (=ei-paikallinen) tai 1 (=paikallinen)

	▪ current-elementti, jos haetaan voimassa olevia koodeja (status), sisältönä päivämäärä, jolloin voimassa olevia haetaan ▪ parentId-elementti, jos haetaan alihierarkiasta (hierarchy) ▪ propertyCodeList-elementti (vapaaehtoinen, oletuksena pelkkä koodiarvo), jossa yksi tai useampia ○ property-elementtejä, sisältönä elementin nimi josta haetaan, attribuuttina tarvittaessa (multilingual) ▪ language-attribuutti, arvon kielen tunniste (jos toteutetaan multilingual-taso freeElementsin lisäksi) howMany-elementti: kerralla palautettavien maksimimäärä (vapaaehtoinen), palvelin palauttaa virheilmoituksen jos osumia on enemmän kuin määritelty tai palvelimen toteutuskohdasta ”liikaa” sortBy-elementti, paluuarvojen lajittelukenttä, vapaaehtoinen, oletuksena koodiarvo, sisältö: ks. liite 3 display-elementti (vapaaehtoinen, ”freeElements”), palautettavat elementit, sisältää ▪ propertyCodeList-elementin, jossa yksi tai useampia ▪ property-elementtejä, sisältönä palautettavan (näytettävän) elementin nimi, attribuuttina tarvittaessa (multilingual) ○ language-attribuutti, arvon kielen tunniste (jos toteutetaan multilingual-taso freeElementsin lisäksi)
Response	Lista joko term- tai termItemEntry-elementtejä. termItemEntry-elementtejä käytetään vain jos toteutetaan freeElements-taso ja hakuparametreissa on määritelty display-elementti. term-elementin sisältönä (pakollinen) koodin näytettävä selite (shortname), attribuuttina ▪ id – koodin tunniste (pakollinen) ▪ language – selitteen kieli (multilingual-tasolla) termItemEntry – elementin ▪ attribuuttina id – koodin tunniste (pakollinen) ▪ sisältönä joukko ▪ attribute-elementtejä (ks. liite 3), joista palautettava ainakin ○ type-attribuutti (ominaisuuden nimi) ○ tietosisältö (näytettävä arvo elementistä) ○ language-attribuutti, jos tuetaan useita kieliä ja palautetaan halutun kielen arvo
Kuvaus	Palvelu palauttaa ne koodit, jotka vastaavat annettuja hakuarvoja ja ominaisuuksia, hakuarvojen pienet ja isot kirjaimet vastaavat toisiaan.
Pakollisuus	Minimitasolla ei vaadittu. Perustasolla ”base” toteutettava, palvelun tunnistettava termSystem ja sen id, find-elementin matchText (ja sen partial=0 ja partial=1 arvot) sekä find-elementin propertyCodeList:issä nimet id ja shortname (oletuksena id, jolloin propertyCodeList voidaan jättää pois). Palvelun tunnistettava myös howMany ja sortBy-elementit (sortBy arvoina id tai shortname, oletuksena id). Monikieliset koodistot ”multilingual”-tasolla palvelun tunnistettava matchTextin language-attribuutti (jota käytetään haussa ominaisuuksissa joihin on määritelty kieli) ja find-elementin propertyjen sisällä oleva language-attribuutti (haetaan vain tietyn kielistä ko. ominaisuudesta). Palvelun on palautettava term:n tai attributen language-attribuutit jos saatavilla. Jos lisäksi toteutettu vapaa tietosisältö, tunnistettava display-elementin property-elementin attribuuttina kieli jos saatavilla. Hierarkkisten koodistojen käsittely ”hierarchy” tasolla palvelun tunnistettava find-elementin parentId-elementti, haettava vain annetun juuren alla olevalla tasolla löytyneitä tuloksia. Paikalliset koodistot ja tila ”status” tasolla palvelun tunnistettava find-elementin status, current ja local elementit ja palautettava niitä vastaavat tulokset, jos sovellettavissa

	Vapaa tietosisältö ”freeElements” tasolla palvelun tunnistettava display elementti ja palautettava termItemEntry-elementtejä term-elementtien sijaan, lisäksi ks. ”multilingual”. Tunnistettava sortBy-elementtinä muutkin tuetut sisältöelementit kuin id ja short-name, jos tuetaan lisäksi ”advSearch”-tasoa. Edistyneet haut ”advSearch” tasolla palvelun on hyväksyttävä useita find-elementtejä ja myös partial=2-hakuarvoja. Taso voidaan toteuttaa ilman freeElements-tasoa tai yhdessä sen kanssa.
Lisätietoja	Jos haetaan vain ”yhden kielisiä” arvoja, find-elementin matchtext-arvon language-attribuuttiin asetetaan haluttu kieli ja jätetään find-elementin propertyjen kielet määrittelemättä. Jos haetaan ”usean kielisiä” ja kielet joista haetaan halutaan eritellä, käytetään find-elementin propertyjen language-ominaisuutta ja jätetään matchtext:in language määrittelemättä. Jos ”advSearch”-tasolla käytetään useita find-elementtejä, palautetaan vain tulokset, jotka täyttävät kaikkien find-elementtien hakuehdot.
Poikkeukset	Paluuarvo on tyhjä response-elementti, jos hakuarvoja vastaavia koodeja ei löydy. TooManyCodes (jos how_many-parametri ylittää koodistotarjoajan toteutuskohtaisen maksimimäärän tai tuloksia löytyy yli määritellyn määrän), GeneralFailure, NotImplemented, MissingParameter, UnknownAttribute, UnknownCodeSystem, UnknownLanguage
Vaatimukset	K2.6

Kutsu DTD

```

<!ELEMENT request (interface,method,param)>
<!ELEMENT param (termSystem,find+,howMany?,sortBy?,display?)>
<!ELEMENT find (matchtext,status?,local?,current?,parentId?,
  propertyCodeList?)>
<!ATTLIST matchtext
  language CDATA
  partial CDATA>
<!ELEMENT display (propertyCodeList)>
<!ELEMENT propertyCodeList (property+)>
<!ATTLIST property
  language CDATA>

```

Kutsu esim.1 (perustaso)

```

<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>LookupCodes</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996/">
      <find>
        <matchText partial="1">B15.</matchText>
      </find>
    </param>
  </request>

```

Vastaus esim. ks. ListCodes.

Kutsu esim.2 (vapaa tietosisältö, status, edistyneet haut, monikieliset, hierarkia) (etsitään selitteistä ja pitkistä nimestä ”maksatul” alkuiset osumat, jotka ovat käytössä eivätkä paikallisia, hierarkian ”A00-A09” alta, lajiteltuna selitteen mukaan, palautetaan tunnisteiden lisäksi pelkkä selite)

```

<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>LookupCodes</method>
  <param>

```

```

<termSystem id="1.2.246.537.6.1.1996"/>
<find>
  <matchText language="fi" partial="1">maksatul</matchText>
  <status>1</status>
  <local>0</local>
  <parentId>B15</parentId>
  <propertyCodeList>
    <property>shortname</property>
    <property>longname</property>
  </propertyCodeList>
</find>
<howMany>30</howMany>
<sortBy>shortname</sortBy>
<display>
  <propertyCodeList>
    <property>shortname</property>
  </propertyCodeList>
</display>
</param>
</request>

```

Vastaus esim.

```

<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <termItemEntry id="B15.0">
    <attribute type="shortname" language="fi">Maksatulehdus A,
      maksakooma</attribute>
  </termItemEntry>
  <termItemEntry id="B15.9">
    <attribute type="shortname" language="fi">Hepatiitti A ilman
      maksakoomaa</attribute>
  </termItemEntry>
</response>

```

5.7.4 IsCodeValid - koodin validointi

Method	IsCodeValid
Interface	Codeset
Käyttötarkoitus	Koodiarvon olemassaolon tarkistaminen tietyssä koodistossa, sovelluksen kentässä olevan koodin oikeellisuuden varmistaminen
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste ▪ version-attribuutissa tarvittaessa koodiston version tarkenne term -elementti, jonka id -attribuutissa validoitava koodiarvo
Response	value -elementti, jonka arvona 0 (ei validi koodiarvo) tai 1 (validi koodiarvo)
Kuvaus	Käytetään, jos halutaan varmistua siitä, löytyykö koodiarvo koodistosta tai sen versios- ta
Pakollisuus	Pakollinen perustasolla ”base”.
Lisätietoja	Ks. myös Code-rajapinnat (koodin voimassaolo ja paikallisuus)
Poikkeukset	GeneralFailure, NotImplemented, MissingParameter, UnknownCodeSystem
Vaatimukset	K2.9

Kutsu DTD

```

<!ELEMENT request (interface,method,param)>
<!ELEMENT param (termSystem,term)>
<!ATTLIST term
  id CDATA #REQUIRED>

```

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>IsCodeValid</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <term id="B15.1"/>
  </param>
</request>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <value>1</value>
</response>
```

5.7.5 GetSupportedCodesetServices - koodistossa tuetut palvelut

Method	GetSupportedCodesetServices
Interface	Codeset
Käyttötarkoitus	Yksittäisen koodiston tukemien palvelutasojen listaus kooditarjoajan toteuttamana
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste
Response	Joukko service -elementtejä, joista kunkin • id-attribuutissa palvelutason tunniste, (pakollinen), jokin seuraavista ”base” – perustaso ”multilingual” – monikieliset koodistot ”freeElements” – vapaa tietosisältö ”advSearch” – edistyneet haut ”hierarchy” – koodistohierarkian käsittely, huom. hierarkkisia koodistoja voidaan käsitellä myös ”ei-hierarkkisina” ”status” – paikallisten koodien ja tilan huomiointi • tekstisisältönä palvelutason näytettävä nimi (vapaaehtoinen, suositellaan) • version-attribuutissa viittaus standardiin, jossa määritellystä palvelutasosta on kyse, tämän dokumentin osalla ”2.0”, myöhemmissä versioissa voidaan käyttää OID-koodia.
Kuvaus	Palvelu palauttaa tiedot siitä, mitä koodistopalveluita se tukee kyseiselle koodistolle.
Pakollisuus	Pakollinen perustasolla ”base”.
Lisätietoja	Koodistotarjoajan ei tarvitse toteuttaa kaikkia esim. kansallisen koodistopalvelimen tietosisällön mahdollistamia tasoja. Toteutuskohtaisesti on dokumentoitava, mitkä tasot toteutus sisältää kustakin koodistosta ja mitkä ovat kunkin koodiston sisältömäärittelyt suhteessa tämän määrittelyn elementteihin. (ks. luku 6)
Poikkeukset	GeneralFailure, NotImplemented, MissingParameter, UnknownCodeSystem
Vaatimukset	K2.8

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>GetSupportedCodesetServices</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
  </param>
</request>
```

Vastaus esim.

ks. GetSupportedServices

5.7.6 GetCodesetInfo - Koodiston hallinnolliset tiedot

Method	GetCodesetInfo
Interface	Codeset
Käyttötarkoitus	Koodiston hallinnollisten tietojen haku
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste
Response	Voi palauttaa tyhjän response-elementin. termSystem -elementti, jonka ▪ sisältönä koodiston nimi ▪ id-attribuutissa koodiston tunniste description -elementti (vapaaehtoinen) Joukko services -elementtejä (vapaaehtoinen, toistuva), ks. GetSupportedServices Joukko language -elementtejä (vapaaehtoinen, toistuva), ks. ListLanguages
Kuvaus	Palautettavasta kuvauksesta voi selvittää mm. koodiston ylläpitäjä, yhteystiedot, osoite johon lähettää kehitysideat, päivitysväli jne.
Pakollisuus	Pakollinen perustasolla "base", voi palauttaa myös tyhjän response-elementin. Languages-lista järkevää toteuttaa vain "multiLingual" ja "freeElements"-tasoilla.
Lisätietoja	
Poikkeukset	GeneralFailure, NotImplemented, MissingParameter, UnknownCodeSystem
Vaatimukset	K2.4

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>GetCodesetInfo</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
  </param>
</request>
```

Vastaus DTD

```
<!ELEMENT response (termSystem,description?,service*,language*)>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <termSystem id="1.2.246.537.6.1.1996">ICD 10 1996</termSystem>
  <description>Stakes, created 1996-10-01</description>
  <service id="base" version="2.0">Perustaso</service>
  <service id="multilingual" version="2.0">Monikieliset</service>
  <service id="hierarchy">Koodistohierarkia</service>
  <language id="fi">Suomi</language>
  <language id="sv">Svenska</language>
  <language id="la">Latin</language>
</response>
```

5.7.7 ListLanguages – koodiston tukemat kielet

Method	ListLanguages
Interface	Codeset
Käyttötarkoitus	Tietyissä koodistossa tuettujen kielten listaus
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste
Response	lista language -elementtejä, • id-attribuutina kielen tunniste (pakollinen) • tietosisältönä kielen näytettävä nimi
Kuvaus	Monikieliset koodistot ”multilingual”-toteutuksissa palvelu voi tarjota koodiarvoja vastaavia selitteitä useilla kielillä niissä koodistoissa, joissa on usean kielisiä selitteitä. Vapaa tietosisältö ”freeElements”-toteutuksissa myös muita elementtejä voi olla usean kielisinä.
Pakollisuus	Pakollinen, jos toteutetaan monikieliset koodistot ”multilingual”-taso.
Lisätietoja	
Poikkeukset	GeneralFailure, NotImplemented, MissingParameter, UnknownCodeSystem
Vaatimukset	K2.12

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>ListLanguages</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
  </param>
</request>
```

Vastaus DTD

```
<!ELEMENT response (language*)>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <language id="fi">Suomi</language>
  <language id="sv">Svenska</language>
  <language id="la">Latin</language>
</response>
```

5.7.8 GetCodes - yksilöityjen koodien halutun sisällön hakeminen

Method	GetCodes
Interface	Codeset
Käyttötarkoitus	Samana koodiston usean yksilöidyn koodin noutaminen halutulla tietosisällöllä
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste lista term -elementtejä, joiden • id-attribuutissa koodin tunniste propertyCodeList -elementti, jossa yksi tai useampia • property-elementtejä, sisältönä palautettavan (näytettävän) elementin nimi, attribuutina tarvittaessa (multilingual) language-attribuutti, kielen tunniste (jos toteutetaan multilingual-taso freeElementsin lisäksi)

Response	lista termItemEntry -elementtejä, • id-attribuuttina koodin tunniste (pakollinen) • sisältönä pyynnössä olleen property-elementin mukaisesti attribute-elementtejä, joissa • sisältönä näytettävä arvo • type-attribuutissa property-parametria vastaava ominaisuuden nimi • language-attribuutissa ominaisuuden kieli pyynnön mukaisesti ”multilingual”
Kuvaus	Halutusta koodijoukosta, esim. haun tuloksena saaduista koodien tunnisteista voidaan tämän operaation avulla hakea halutut kentät näytettäväksi (ilman toistuvia hakuja tai koodikohtaista tietojen hakua).
Pakollisuus	Pakollinen, jos toteutetaan vapaa tietosisältö ”freeElements”-taso. Ks. myös language-attribuuttien käyttö jos lisäksi ”multilingual” taso toteutettu.
Lisätietoja	
Poikkeukset	UnknownAttribute (jos koodiston tietokentässä ei ole jotain nimetyistä kentistä), TooManyCodes (toteutuskohtaisesti voidaan määritellä maksimimäärä sallittuja code_id-arvoja), GeneralFailure, NotImplemented, MissingParameter, UnknownConceptCode, UnknownCodeSystem, UnknownLanguage
Vaatimukset	K4.3

Kutsu DTD

```
<!ELEMENT request (interface,method,param)>
<!ELEMENT param (termSystem,term+,propertyCodeList?)>
```

Kutsu esim

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>GetCodes</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <term id="D68.9"/>
    <term id="D69.8"/>
    <propertyCodeList>
      <property>longname</property>
      <property language="sv">shortname</property>
    </propertyCodeList>
  </param>
</request>
```

Vastaus DTD

```
<!ELEMENT response (termItemEntry*)>
```

Vastaus esim

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <termItemEntry id="D68.9">
    <attribute type="longname" language="fi">Määrittämätön
      hyytymishäiriö</attribute>
    <attribute type="shortname" language="sv"> Koagulationsrubbning,
      ospecificerat</attribute>
  </termItemEntry>
  <termItemEntry id="D69.8">
    <attribute type="longname" language="fi">Muu
      verenvuototila</attribute>
    <attribute type="shortname" language="sv"> Annat specificerat
      hemorragiskt tillstånd</attribute>
  </termItemEntry>
</response>
```

5.7.9 GetSupportedAttributes - koodiston tietokenttien kysely

Method	GetSupportedAttributes
Interface	Codeset
Käyttötarkoitus	Koodiston sisältämien sisältöelementtien nimien hakeminen
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste
Response	propertyCodeList -elementti, joka sisältää • property-elementtejä, joiden ○ sisältönä ominaisuuden nimi ○ language-attribuutissa ominaisuuden kielen tunniste (jos "multilingual")
Kuvaus	Hyödyntäjä kutsuu operaatiota, kun se tarvitsee tiedon siitä, mitä sisältöelementtejä tietty koodisto kaiken kaikkiaan sisältää.
Pakollisuus	Pakollinen, jos toteutetaan vapaa tietosisältö "freeElements"-taso. Ks. myös language-attribuuttien käyttö jos lisäksi "multilingual" taso toteutettu.
Lisätietoja	Sama ominaisuus voidaan palauttaa "useita kertoja" vastauksessa, jos se on saatavilla usealla kielellä "multilingual".
Poikkeukset	GeneralFailure, NotImplemented, MissingParameter, UnknownCodeSystem
Vaatimukset	K2.1

Kutsu DTD

```
<!ELEMENT request (interface,method,param)>
<!ELEMENT param (termSystem)>
```

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>GetSupportedAttributes</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
  </param>
</request>
```

Vastaus DTD

```
<!ELEMENT response (propertyCodeList)>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <propertyCodeList>
    <property>shortname</property>
    <property>longname</property>
    <property>hierarchyLevel</property>
    <property>parentId</property>
  </propertyCodeList>
</response>
```

5.7.10 GetHierarchyDepth – koodistohierarkian syvyys

Method	GetHierarchyDepth
Interface	Codeset
Käyttötarkoitus	Koodistohierarkian syvyyden kysely
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste parentId -elementti, jonka sisältönä hierarkian juuren tunniste (jos elementti puuttuu, palautuu syvyys koodiston juuresta)
Response	value -elementti, jonka sisältönä hierarkian syvyys
Kuvaus	Palauttaa juuresta tai halutusta solmusta laskien hierarkian syvyyden. 0=ei alitasoja), yksitasoisen koodiston juuresta (ilman parametria) 1. Palautettava syvyys on hierarkian maksimisyvyys (syvin alihierarkia) halutusta solmusta.
Pakollisuus	Pakollinen, jos toteutetaan hierarkkisten koodien käsittely ”hierarchy” -taso
Lisätietoja	Code-rajapinnassa on myös GetHierarchyLevel-operaatio, jolla saadaan yksittäisen koodin hierarkiatasto.
Poikkeukset	GeneralFailure, NotImplemented, MissingParameter, UnknownConceptCode (jos parentId virheellinen), UnknownCodeSystem
Vaatimukset	

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Codeset</interface>
  <method>GetHierarchyDepth</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <parentId>D70</parentId>
  </param>
</request>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <value>1</value>
</response>
```

5.8 Code-rajapinta

Koodi yksilöidään Code-rajapinnassa term- tai termItemEntry-elementin id-attribuutilla. Koodisto yksilöidään Code-rajapinnassa samoin kuin Codeset-rajapinnassa termSystem-elementin id-attribuutilla.

5.8.1 GetDesignation - koodin arvon hakeminen

Taulukko 5.3. Code-rajapinnan operaatiot.

Method	GetDesignation
Interface	Code
Käyttötarkoitus	Selitteen hakeminen sovellukseen tietyllä koodiarvolla
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste term -elementti, jonka ▪ id-attribuutissa koodin tunniste ▪ language-attribuutissa halutun selitteen kielen tunniste
Response	term -elementti, jonka • sisältönä näytettävä selite (pakollinen) • id-attribuutin arvona koodin tunniste (pakollinen) • language-attribuutin arvona kielen tunniste
Kuvaus	Hyödyntäjä kutsuu operaatiota, kun se tarvitsee koodiarvon perusteella selitteen yhdestä koodista.
Pakollisuus	Pakollinen minimitasolla. ”multilingual”-tasolla tuettava term-elementin language-attribuutin käyttöä kutsussa ja paluuarvossa.
Lisätietoja	
Poikkeukset	UnknownConceptCode (jos id:llä ei löydy koodia), GeneralFailure, NotImplemented (vain jos ei tueta language-attribuuttia tai kutsussa ”ylimääräisiä attribuutteja”), MissingParameter, UnknownCodeSystem, UnknownLanguage
Vaativuudet	K4.1

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Code</interface>
  <method>GetDesignation</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <term id="G24.5" language="se"/>
  </param>
</request>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <term id="G24.5" language="se">Blefaroospasm</term>
</response>
```

5.8.2 GetParent – ylemmän tason koodi

Method	GetParent
Interface	Code
Käyttötarkoitus	Hierarkiassa ylemmän tason elementin kysely
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste term -elementti, jonka ▪ id-attribuutissa koodin tunniste ▪ language-attribuutissa pyydetty selitteen kieli ”multilingual”

Response	term -elementti, jonka - sisältönä (ylemmän tason koodin) näytettävä selite id-attribuutin arvona koodin tunniste (pakollinen) - language-attribuutin arvona selitteen kielen tunniste (vapaaehtoinen, ”multilingual”)
Kuvaus	Palauttaa halutun elementin vanhemman (hierarkiassa ylemmällä tasolla olevan koodin). Jos hierarkiassa ei ole ylempiä tasoja, palautuu virhe.
Pakollisuus	Pakollinen, jos toteutetaan hierarkkisten koodien käsittely ”hierarchy”-taso
Lisätietoja	Code-rajapinnassa on myös GetHierarchyLevel-operaatio, jolla saadaan yksittäisen koodin hierarkiataso.
Poikkeukset	GeneralFailure, NotImplemented, MissingParameter, UnknownConceptCode, UnknownCodeSystem, UnknownLanguage
Vaatimukset	K2.10

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Code</interface>
  <method>GetParent</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <term id="G24.5"/>
  </param>
</request>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <term id="G24">Lihaskäntäjäshäiriö</term>
</response>
```

5.8.3 GetStatus - koodin tila

Method	GetStatus
Interface	Code
Käyttötarkoitus	Koodin tilan kysely
Param	termSystem -elementti, jonka id-attribuutissa koodiston tunniste term -elementti, jonka id-attribuutissa koodin tunniste
Response	value -elementti, jonka arvona 1 (=active, käytössä), 2 (=delete, poistettu) tai 0 (=work, ehdotettu)
Kuvaus	Palauttaa kysytyn koodin tilatiedon, oletuksena active
Pakollisuus	Pakollinen, jos toteutetaan koodien paikallisuus ja tila ”status”-taso
Lisätietoja	
Poikkeukset	UnknownConceptCode, GeneralFailure, NotImplemented, MissingParameter, UnknownConceptCode, UnknownCodeSystem
Vaatimukset	K2.15

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Code</interface>
  <method>GetStatus</method>
  <param>
```

```

    <termSystem id="1.2.246.537.6.1.1996"/>
    <term id="G24.5"/>
  </param>
</request>

```

Vastaus esim.

```

<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <value>1</value>
</response>

```

5.8.4 GetLocal - koodin paikallisuus

Method	GetLocal
Interface	Code
Käyttötarkoitus	Koodin paikallisuuden kysely
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste term -elementti, jonka ▪ id-attribuutissa koodin tunniste
Response	value -elementti, jonka arvona 1 (=paikallinen) tai 0 (=ei-paikallinen)
Kuvaus	Palauttaa kysytyn koodin paikallisuustiedon, oletuksena 0
Pakollisuus	Pakollinen, jos toteutetaan koodien paikallisuus ja tila "status"-taso
Lisätietoja	Määrittäminen ei ota kantaa, onko koodi lisätty palveluun juuri sillä tasolla, jolla palvelu toimii vai jollain ylemmällä tasolla (esim. lisäys voi tapahtua alueellisella tasolla ja palvelu sijaita organisaatiotasolla, jolloin get_local voisi palauttaa periaatteessa kumman arvon tahansa). Tämä on syytä tarkentaa toteutuskohtaisesti.
Poikkeukset	UnknownConceptCode, NotImplemented, GeneralFailure, MissingParameter, UnknownCodeSystem
Vaatimukset	K2.15

Kutsu esim.

```

<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Code</interface>
  <method>GetLocal</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <term id="G24.5"/>
  </param>
</request>

```

Vastaus esim.

```

<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <value>0</value>
</response>

```

5.8.5 LookupCompleteCodedConcept - koodin koko sisällön hakeminen

Method	LookupCompleteCodedConcept
Interface	Code
Käyttötarkoitus	Kaikkien koodin tietojen hakeminen sovellukseen tietyllä koodilla
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste term -elementti, jonka ▪ id-attribuutissa koodin tunniste
Response	termItemEntry -elementti, jossa • id-attribuuttina koodin tunniste • joukko attribute-elementtejä, joissa kussakin yksi koodin ominaisuus (tietokenttä), kullakin attribute-elementillä on ○ tietosisältönä näytettävä arvo (pakollinen) ○ type-attribuutin arvona tietokentän nimi (pakollinen) ○ mahd. language-attribuutti, (ks. liite 3) ○ mahd. muita attribuutteja (ks. liite 3)
Kuvaus	Hyödyntäjä kutsuu operaatiota, kun se tarvitsee koodiarvon perusteella kaiken sisällön yhdestä koodista.
Pakollisuus	Pakollinen, jos toteutetaan vapaa tietosisältö ”freeElements”-taso
Lisätietoja	Jos annetulla koodiarvolla ei löydy koodia, paluuarvo on tyhjä.
Poikkeukset	GeneralFailure, NotImplemented, MissingParameter, UnknownConceptCode, UnknownCodeSystem, UnknownLanguage
Vaatimukset	K4.2

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Code</interface>
  <method>LookupCompleteCodedConcept</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <term id="G24.5"/>
  </param>
</request>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <termItemEntry id="G24.5">
    <attribute type="shortname" language="fi">
      Luomikouristus</attribute>
    <attribute type="hierarchylevel">3</attribute>
    <attribute type="parentId">G24</attribute>
    <attribute type="longname">Luomikouristus</attribute>
    <attribute type="beginDate">1995-20-06</attribute>
    <attribute type="expirationDate">2020-12-31</attribute>
    <attribute type="lastModifiedDate">2004-02-16</attribute>
    <attribute type="lastModifiedBy">Stakes, Admin</attribute>
    <attribute type="status">1</attribute>
    <attribute type="local">0</attribute>
    <attribute type="shortname" language="sv">
      Blefarospasm</attribute>
    <attribute type="shortname" language="la">
      Blepharospasmus</attribute>
  </termItemEntry>
</response>
```

5.8.6 LookupProperties - koodin halutun sisällön hakeminen

Method	LookupProperties
Interface	Code
Käyttötarkoitus	Yhden koodin haluttujen sisältöelementin hakeminen sovellukseen
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste term -elementti, jonka ▪ id-attribuutissa koodin tunniste propertyCodeList -elementti, jonka • property-elementeissä on ◦ sisältönä ominaisuuden nimi (pakollinen) ◦ language-attribuutissa ominaisuuden halutun kielen tunniste ("multi-lingual")
Response	termItemEntry -elementti, jossa • id-attribuuttina koodin tunniste • joukko attribute-elementtejä, joissa kussakin yksi (pyydettyä propertyä vastaava) koodin ominaisuus (tietokenttä), kullakin attribute-elementillä on • tietosisältönä näytettävä arvo (pakollinen) • type-attribuutin arvona tietokentän nimi (pakollinen) • mahd. language-attribuutti (pakollinen jos pyynnössä oli language-attribuutti)
Kuvaus	Hyödyntäjä kutsuu operaatiota, kun se tarvitsee koodiarvon perusteella määrättyjen sisältökenttien arvon yhdestä koodista.
Pakollisuus	Pakollinen, jos toteutetaan vapaa tietosisältö "freeElements"-taso
Lisätietoja	
Poikkeukset	UnknownConceptCode, UnknownAttribute, GeneralFailure, NotImplemented, Missing-Parameter, UnknownCodeSystem, UnknownLanguage
Vaativuudet	K4.3

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Code</interface>
  <method>LookupProperties</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <term id="G24.5"/>
    <propertyCodeList>
      <property>shortname</property>
      <property>hierarchyLevel</property>
      <property>parentId</property>
    </propertyCodeList>
  </param>
</request>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <termItemEntry id="G24.5">
    <attribute type="shortname">Luomikouristus</attribute>
    <attribute type="hierarchylevel">3</attribute>
    <attribute type="parentId">G24</attribute>
  </termItemEntry>
</response>
```

5.8.7 GetHierarchyLevel - koodin hierarkiataso

Method	GetHierarchyLevel
Interface	Code
Käyttötarkoitus	Koodin hierarkiatason kysely
Param	termSystem -elementti, jonka ▪ id-attribuutissa koodiston tunniste term -elementti, jonka ▪ id-attribuutissa koodin tunniste
Response	value -elementti, jonka arvona hierarkiataso (0=ylin)
Kuvaus	Palauttaa kysytyn koodin hierarkiatason
Pakollisuus	Pakollinen, jos toteutetaan koodien hierarkkisten koodien käsittely ”hierarchy”-taso
Lisätietoja	
Poikkeukset	UnknownConceptCode, GeneralFailure, NotImplemented, MissingParameter, UnknownCodeSystem
Vaatimukset	K2.10

Kutsu esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns="urn:plugit:CommonServices">
  <interface>Code</interface>
  <method>GetHierarchyLevel</method>
  <param>
    <termSystem id="1.2.246.537.6.1.1996"/>
    <term id="G24.5"/>
  </param>
</request>
```

Vastaus esim.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <value>3</value>
</response>
```

5.9 Virhetilanteet

Eri operaatioiden kuvauksissa on lueteltu virheet, jotka operaatio voi palauttaa. Esimerkki virhetilanteesta tulevasta vastauksesta palvelulta:

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:plugit:CommonServices">
  <exception id="GeneralFailure">Odottamaton virhe</exception>
</response>
```

Seuraava taulukko sisältää tarkat virhekoodit ja ohjeelliset virheilmoitukset tässä dokumenteissa määritellyille virheille.

Taulukko 5.4. Koodipalvelun virhekoodit ja niiden merkitykset.

Koodi	Ilmoitus esim.	Merkitys
GeneralFailure	Yleinen virhe	(vast. esim. UnexpectedError (Solbrig 2003))
NotImplemented	Operaatiota tai parametrina saatua elementtiä ei ole toteutettu: (nimi)	Käytetään esim. tilanteissa, joissa parametrina tulee elementti, attribuutti tai attribuutin arvo, joka on määritelty tässä dokumentissa, mutta jota palvelu ei tue (esim. palvelu tukee vain perustasoa, mutta parametrina tulee kieli tai partial=2-arvo), tai rajapinta tai operaatio jota ei tueta. Voidaan käyttää myös jos elementti tai attribuutti sisältää tietoa jota palvelin ei ymmärrä, (mutta voidaan myös hyväksyä laajennukset).
MissingParameter	Pakollinen tieto puuttuu: (parametrin nimi)	Jokin jokin pakollisista tai operaatioon vaadituista tiedoista (esim. elementti, attribuutti tai sisältö) puuttuu kutsun parametreista.
TooManyCodes	Liian monta pyydettyä koodia	howMany-parametri on liian suuri tai haku palauttaa (tai palauttaisi) liian paljon tuloksia.
UnknownAttribute	Nimetty tieto ei sisälly koodistoon	Esim. pyydetään tai lajittelussa käytetään tietoelementtiä, joka ei sisälly koodistoon (sen sisältöomäilytykseen). Liittyy ”freeElements” tasolla property ja attribute type-sisältöön, muilla tasoilla voi liittyä sortBy-elementtiin. Jos tämän määrittelyn status- ja local-elementtejä ei tueta, palautetaan tämän sijaan NotImplemented.
UnknownConceptCode	Tuntematon koodi	Koodin tunnisteella ei löydy koodistosta koodia(CTS)
UnknownCodeSystem	Tuntematon koodisto	Koodiston tunnisteella ei löydy koodistoa (CTS)
UnknownLanguage	Tuntematon kieli	Kielen tunnisteella ei löydy kieltä tai kieli ei ole tuettu (CTS)

5.10 Laajennukset ja jatkokehitys

Eri koodistot liitetään rajapintaan sisältöomäilytyksen (luku 6) avulla. Sisältöomäilytyksiä voidaan lisätä ilman, että rajapintamäärittelystä tehdään uusia versioita.

Versiossa 2.0 esiin tulleita jatkokehitysideoita:

Hierarkkisissa koodistoissa tehtävissä hauissa haku voitaisiin ulottaa lehtitasolle (vain juurta seuraavan tason lisäksi) esim.lisäämällä parentId-elementtiin sopiva attribuutti.

Arvojoukkojen rajaukset voitaisiin toteuttaa lisäämällä eri kutsuihin vocabularydomain-attribuutti, jolla arvojoukko rajattaisiin ennalta määriteltyyn yksilöityyn osajoukkoon (esim. kuntakoodistosta ”HUS-alueen kunnat”). Tätä varten tarvittaisiin esim. seuraavat operaatiot ja muutokset:

- Codeset:GetSupportedVocabularyDomains
- Codeset:LookupCodesByDesignation: hyväksyttävä vocabularyDomain-parametri, palautettava vain tämän rajauksen mukaiset tulokset
- Codeset:LookupCodes: ks. edell.
- Codeset:ListCodes: ks. edell.

Koodiarvojen ja koodistojen suhteita voitaisiin määritellä, kun otettaisiin käyttöön relationship-käsite ja sitä vastaavat elementit. Tämä mahdollistaisi mm. operaatiot:

- Codeset:GetSupportedRelationships – koodistossa tuetut suhteet
- Codeset:AreCodesRelated – koodiarvojen suhteiden kysely

- Codeset:TranslateConceptCode – koodien muunnokset koodistojen välillä
- Esimerkiksi kysely ”HUS-alueen kunnat” voitaisiin siis periaatteessa toteuttaa sekä arvojoukkojen että koodistojen välisten suhteiden avulla.

Koodistojen voimassaolopäivämäärät voidaan lisätä esim. palautettaviin tietoihin GetCodeSetInfo-operaatioissa, jos tarvetta ilmenee. Oletusarvoisesti pyritään käyttämään koodiston uusinta versiota.

5.11 Toteutuksiin kohdistuvia vaatimuksia

5.11.1 Toteutuskohtaisesti ratkaistavat seikat

Luvun 5 teknisessä määrittelyssä ei ole määritelty:

- Mitkä koodistot palvelu tarjoaa (esim. ICD-10, organisaatiokoodisto jne.). Stakesin koodistopalvelimelle suunnitelluista koodistoista on lista dokumentissa (Koodistopalvelu 2004) sekä tämän dokumentin luvussa 2.5. Myös raportissa (Mykkänen ym. 2004b) on lueteltu keskeisiä koodistostandardeja ja annettu osoitetta niiden ylläpitäjätahoihin. Kustakin palvelun tarjoamasta koodistosta on syytä käyttää avointa ja eri osapuolten saatavilla olevaa sisältömäärittystä (luku 6).
- Vaatiiko koodistotarjoaja koodistohyödyntäjä-sovelluksen tai käyttäjän yksilöintiä ja kuinka tämä yksilöinti ja sen varmistaminen tapahtuu. Tarvittaessa tunnistus ja yksilöinti voidaan toteuttaa esim. PlugIT-projektin käyttäjä- ja oikeus-rajapintojen määrittysten mukaisesti (Sormunen ym. 2004b).
- Vastuita siinä tapauksessa, että koodistotarjoaja on esim. teknisen vian vuoksi saavuttamattomissa. Tämä ja vastaavat tilanteet sovitaan tuotekohtaisesti, mutta hyödyntäjäohjelmistoissa on syytä varautua teknisiin ja verkkovirheisiin.

Näitä seikkoja tulisi kuvata tapaus- ja toteutuskohtaisesti mm. tuotedokumentaatiossa ja tarjouspyynnöissä.

Eri koodistojen sisältömäärittelyyn viittamalla ja niitä tarkentamalla voidaan kuvata:

- Jos samassa koodistossa yhdellä (varsinaisella) koodiarvolla on useita selitteitä, on dokumentoitava koodistokohtaisesti, käytetäänkö yksilöivänä tunnisteena (koodiarvo) jotain muuta tietoa kuin varsinainen koodiarvo (esim. ”rivinumero”). Tunnisteeksi suositellaan samaa tunnistetta kuin Stakesin koodistopalvelu-määrittelyissä. Tällöin on myös kuvattava, vastaako esim. peruslistaus-operaatioiden ”selite/value” koodiarvoa vai selitettä, ja käytetäänkö esim. freeElements-tason operaatioita varsinaisen selitteen tai koodiarvon hakuun. Suositus on käyttää tällöin value-arvona ”koodiarvoa” ja noutaa selitteet tarvittaessa esim. freeElements-mekanismeilla.
- mitkä ovat freeElements-tasolla tuetut sisältöelementit (property) kussakin koodistossa,
- jos status-taso on toteutettu, tuetaanko koodiston osalta kaikkia rajoitteita,
- ks. myös luvut 4.5 ja 7.

5.11.2 Määrittelyn mukaisuudesta varmistuminen

Tämän määrittelyn mukaiset toteutukset toimivat luvun 5.3 teknisten periaatteiden mukaisesti:

- http-kutsuissa käytetään POST-operaatiota
- Siirrettävän tiedon http content-type on text/xml sekä pyynnöissä että vastauksissa
- Merkistökoodauksena XML-sisällössä on UTF-8
- Pyyntöissä ja vastauksissa XML-elementtien järjestys on tämän dokumentin mukainen

- XML-elementtien attribuuttien järjestys on vapaa
- XML-elementtien ja attribuuttien nimien kirjainkoot ovat tämän dokumentin mukaisia, samoin elementtien ja attribuuttien sisällössä olevat operaatioiden, rajapintojen, määrittelyn tasojen ja tietoelementtien (esim. property-elementin sisältö) nimet ja kirjainkoot.
- Koodistohyödyntäjä jättää tulkitsematta koodistotarjoajan vastauksissa tulevat mahdolliset ”ylimääräiset” elementit ja attribuutit. Koodistotarjoaja voi palauttaa virheilmoituksen koodistohyödyntäjän lähettämistä ylimääräisistä elementeistä ja attribuuteista. Hyödyntäjä voi käyttää (perustason ja ylempien tasojen) palvelimien kanssa operaatioita, joilla se saa selville tuetut operaatiot ja elementit.

Määrittelyn eri tasojen vaatimukset toteutuksille on kuvattu luvun 5 operaatiokuvauksissa ja koostusti liitteessä 3.

Minimitaso vaaditaan kaikilta toteutuksilta. Perustaso on suositeltava minimitason lisäksi. Muut tasot voidaan toteuttaa niiden lisäksi toisistaan riippumatta. Tasoja ja operaatioita eri tasoille voidaan uusissa versioissa lisätä tarpeen mukaan esim. eri terminologioiden suhteita tai yhdistelmäkäsitteiden käsittelyä varten.

Toteutuksiin voidaan määritellä (ja dokumentoida) laajennuksia, joita ei ole määritelty tässä dokumentissa. Ne eivät kuitenkaan saa olla päällekkäisiä tai haitata tämän dokumentin mukaisia toteutuksia.

Toteutus on tämän dokumentin mukainen, jos se toteuttaa minimitason. Toteutukset voivat toteuttaa suositeltavien minimi- ja perustason lisäksi useita muita tasoja (toisistaan riippumatta). Mm. monikieliset koodistot, koodistohierarkian käsittely ja koodien paikallisuus ja tila-tasot on nähty yleisimpinä koodistotarjoajaan kohdistuvina tarpeina saaduissa kommentteissa. Tarjouspyynnöissä ja integrointiratkaisuissa voidaan vaatia tiettyjen tai kaikkien tasojen toteuttamista.

Toteutukset, jotka noudattavat tämän dokumentin operaatio- ja sisältömäärittelyksiä, teknisiä periaatteita ja määriteltyjä sovellusten vastuita, ovat määrittelyn mukaisia. Määrittelyn mukaisesta toteutuksesta on mainittava, mitkä palvelutasot se toteuttaa ja mitä sisältömäärittelyksiä se noudattaa.

Palvelun toteutuksen määrittelyn mukaisuus voidaan todeta testausraportilla, joka sisältää palvelun toteutuksen palauttamien arvojen ja toiminnan kuvauksen yllä kuvatuilla ehdoilla ja lisäksi joillakin virheellisillä ja puuttuvilla syötteillä sekä tarkastamalla että toteutuskohdaksi jätetyt seikat on ratkaistu ja dokumentoitu.

Palveluja käyttävän sovelluksen määrittelyn mukaisuus voidaan todeta testausraportilla, jossa palvelun onnistuneiden kutsujen (yllä olevilla edellytyksillä) lisäksi selviää palvelua käyttävän sovelluksen toiminta palvelun palauttaessa virhetilanteita tai ollessa saavuttamattomissa. Määrittelyjen mukaisuuden testausta on käsitelty raportissa (Mykkänen ym. 2004d).

6 SISÄLTÖMÄÄRITYKSET: RAJAPINNAN LIITTÄMINEN ERI KOODISTOIHIN

Kun luvussa 5 kuvattu rajapinta liitetään minkä tahansa koodiston tietosisältöön, on kuvattava:

- mikä elementti vastaa id-parametria, ja jos id-arvot eivät ole yksikäsitteisiä (samaa id:tä vastaa usea selite), miten vastataan päällekkäisiin, esim.
 - käytetään vain ensimmäistä löytyvää selitettä,
 - lisätään id-arvoon esim. yksilöivä numero, jolla samasta id-arvosta muodostetaan ”useita”,
 - käytetään useita id-arvoja, joilla yhdessä yksilöidään koodi.
- mikä elementti vastaa selitettä (tässä määrittelyssä selite esim. lajittelukentän sortBy-parametrissa tunnustetaan nimellä shortname, vaikka selite tulisi muusta kentästä kuin ”shortname” ulkoisessa sisältö-määrittelyssä (myös designation, value, abbreviation jne.)
- voiko palvelun toteuttava sovellus tarjota multilingual, hierarchy, status tai freeElements-tasoja, ja mitkä ovat sisällössä näihin tasoihin liittyvät tietokentät tai elementit
 - mitkä kielet sisältyvät viralliseen määrittelyyn, voiko kielen määrittellä selitteen lisäksi muille kentille freeElements-tasolla
 - millainen hierarkia sisältyy viralliseen määrittelyyn
 - onko koodistossa tila- ja paikallisuustietoja
 - mitkä elementit vähintään ovat mukana freeElements-tasossa
 - voidaanko koodiston osalta tietyllä tasolla toteuttaa vain jotkin piirteet (esim. status-tasolla koodien tila-status ja voimassaolo-current, muttei paikallisuutta-local).

Nämä seikat voidaan kuvata rajapinnoista erillisinä *sisältö-määrittelyinä* (content mappings). Sisältö-määrittelysten elementtien nimeämisen pohjana (etenkin freeElements-tasolla) käytetään oletusarvoisesti Stakesin koodistopalvelun ”Koodistojen siirron ja kyselyn XML-rajapinta, versio 1.0” -dokumentin mukaisia attribute type –attribuuttien arvoja (Koodistopalvelu 2004, s. 14-16) ja niihin pohjautuvia koodistojen siirtosanomiam. Kyseisen dokumentin aiemmissa versioissa esiintyneet kirjoitusvirheet (mm. ”atribute”) on korjattu tässä dokumentissa ja myös Koodistopalvelun XML-rajapinnan uudessa versiossa.

6.1 ICD-10 sisältö-määrittely

Alle on koottu esimerkkinä sisältö-määrittely ICD-10-koodistolle. Kyseisen sisältö-määrittelyn avulla tämän dokumentin rajapinnat liitetään Open CDA-määrittelydokumentin (Koodistopalvelu 2004) mukaisen siirtosanomien sisältöelementteihin ja Stakesin koodistopalvelun Excel-siirtotiedoston sarakkeisiin.

Taulukko 6.1. ICD-10-sisältömäärittäminen.

Koodiston nimi	Stakes - Tautiluokitus ICD-10
Koodiston ja version tunnistet	1.2.246.537.6.1 1.2.246.537.6.1.1996 1.2.246.537.6.1.1999.1 (oletus)
Koodiston ylläpitäjä	Stakes 16.02.2004
Sisältömäärittäminen	Juha Mykkänen, Kuopion yliopisto, HIS-yksikkö, 16.4.2004
Sisältömäärittämisestä lähteet	Siirtosanoma = Open CDA-määrittelydokumentti (Koodistopalvelu 2004), Excel = Koodistopalvelu, Stakes, ICD 10 1999-koodisto (Excel-muoto)
Toteutuksissa mahdolliset palvelutasot	minimitaso, base, multilingual (fi-suomi, sv-ruotsi, la-latina), hierarchy, status (ainakin current-elementin osalta), freeElements, advSearch

Minimitason elementti	Lähde, malli, esimerkki
termSystem id	1.2.246.537.6 (siirtosanoma: termSystem id)
termSystem version	1.2.246.537.6.1.1999.1 (oletus)
term id	Siirtosanoma: termItemEntry id Excel: Tunniste
term sisältö, shortname	Siirtosanoma: attribute type="shortname" language="fi" -elementin sisältö Excel: Lyhenne

Monikieliset koodistot (multilingual) elementti	Lähde, malli, esimerkki
term language="fi"	Siirtosanoma: attribute type="shortname" language="fi" elementin sisältö Excel: Lyhenne
term language="sv"	Siirtosanoma: attribute type="shortname" language="sv" elementin sisältö Excel: Ruotsi
term language="la"	Siirtosanoma: attribute type="shortname" language="la" elementin sisältö Excel: Latina

Koodistohierarkia (hierarchy) elementti	Lähde, malli, esimerkki
parentId-elementti (haku-parametrina)	Siirtosanoma: attribute type="parentId" Excel: Ylempi koodi
GetHierarchyLevel	Siirtosanoma: attribute type="hierarchyLevel" Excel: Hierarkiataso

Koodien paikallisuus ja tila (status) elementti	Lähde, malli, esimerkki
status-elementti	Siirtosanoma: attribute type="status" Excel: Tila
local-elementti	Siirtosanoma: attribute type="local" Excel: ei vastinetta
current-elementti	Siirtosanoma: attribute type="beginDate", attribute type="expirationDate" Excel: Voimassaolo alkaa, Voimassaolo päättyy

Vapaa tietosisältö	Lähde, malli, esimerkki
(freeElements) elementti (tuetut property-elementin sisällöt)	id, shortname, Lisäksi: (Siirtosanoma esimerkistä): hierarchyLevel, (hierarkiataso) parentId, (ylemmän tason koodi) beginDate, (voimassaolo alkaa) expirationDate, (voimassaolo päättyy) lastModifiedDate, (viim. muutospäivämäärä) Excel: longname / longname language="fi" (Pitkä nimi), lastModifiedBy (Viim tallentaja), status (Tila), definition (Määritelmä), shortname language="sv" (Ruotsi), shortname language="la" (Latina), (Koodi1, Yhdistelymerkki, Koodi2), leaf (Lehtisolmu)

7 TOTEUTUKSET JA JATKOKEHITYS

Raportissa (Mykkänen ym. 2004d) on kuvattu sitä, kuinka avoimia integrointiratkaisuja ja erityisesti PlugIT-hankkeen rajapintamäärittelyksiä voidaan hyödyntää ohjelmistototeutuksissa ja kuinka toteutusten määrittelyjen mukaisuutta voidaan testata. Raportti sisältää myös toteutuskokemuksia ja esimerkkejä integrointiratkaisuista, joissa on käytetty välineitä ja tekniikoita, jotka soveltuvat myös tämän dokumentin rajapintojen toteuttamiseen. PlugIT-hankkeen taustamateriaaliksi on toteutettu tämän dokumentin Teknisten liittymämäärittelyjen pohjalta Delphi-välineillä Windows-alustalla toimiva koodistopalvelun (koodistotarjoaja) referenssitoteutus ja hankkeessa .NET-välineillä toteutettu koodistohyödyntäjäsovellus. Tarpeita ja halukkuutta rajapintojen toteutuksiin on nähty sekä organisaatioiden ydinjärjestelmissä että osana alueellisia tietojärjestelmiä.

Toteutuksesta tulisi dokumentoida ja tuottaa toteutuksen hyödyntäjien saataville yleisten toteutuskohdainten käytäntöjen dokumentoinnin (Mykkänen ym. 2004c) lisäksi mm.:

- poikkeamat tai tarkennetut tulkinnot tähän määrittelykseen,
- mitkä koodistot ja niiden versiot palvelu tarjoaa,
- mitkä tasot määrittelyksestä on toteutettu palveluun ja eri koodistoihin (ks. liite 2).
- mikä on hakuoperaatioissa kerralla palautettavien vastausten määrän suurin sallittu arvo ja oletusarvo (jos hyödyntäjä ei määrittele `how_many-parametreja`),
- mikä on suurin sallittu määrä kerralla pyydettyjä nimettyjä koodiarvoja,
- mikä on oletuskieli monikielisissä koodistoissa,
- jos palvelussa on toteutettu sekä `freeElements` että `advSearch`-tasot, tukeeko palvelu koodien (tämän dokumentin mukaisesti) lajittelua minkä tahansa kentän mukaan, tai rajoitukset lajittelukenttien suhteen
- asetetaanko koodistohyödyntäjälle lisärajoitteita tai -vaatimuksia esim. koodistojen massiivisen toistuvan listauksen tai muiden koodistotarjoajaa runsaasti kuormittavien operaatioiden suhteen.

Lukujen 4 ja 5 viimeisissä aliluvuissa on lueteltu toteutuskohdaisesti ratkaistavia ja dokumentoitavia asioita, joita määrittelysten mukaisten toteutusten tulee noudattaa.

Uusia versioita ja työstettäviä vaatimus- ja liittymämäärittelyksiä on PlugIT-projektin aikana hyväksytty ja julkistettu projektin hyväksymiskäytäntöjen mukaisesti (Mykkänen ym. 2004c). Projektin päättymisen jälkeen niiden jatkotyöstöä tehdään tarpeen mukaan Suomen HL7-yhdistyksen Common Services SIG-työryhmässä, ja myös jatko- ja yhteistyöstä kansainvälisen HL7:n Common services -ryhmän sekä HL7 CTS-määrittelijöiden kanssa ollaan käymässä keskusteluja.

8 LÄHTEET

- 3M 2002. 3M: Making Sense of the Data: Using a Medical Data Dictionary to Integrate, Share, and Understand Clinical Data. White paper 09_02 (pdf), 2002, 23.9.2003. <http://www.3m.com>.
- CEN 2003. Health Informatics - Data Types. CEN TC251, version 0.2, 16.1.2003.
- Duke JR, Crawford J. Terminology Services. Health Care Information Consultants, LLC, 2001.
- Ensio A, Ruotsalainen P. Tietoturvallinen kommunikaatioalusta: Suositus kansallisesti noudatettaviksi standardeiksi. Osaavien keskustien verkoston julkaisuja 7/2004, Stakes, 2004.
- HL7 2002. HL7 CDA paikallistamisprojekti, soveltamisopas. HL7 Finland, 2002.
- Isokorpi P, Taberman H, Virtanen A, Itälä T. ATJ:n koodistovastaavuudet. Versio 1.2. Satakunnan Makropilotti, 27.3.2001.
- Koodistopalvelu 2003. Koodistojen siirron ja kyselyn Xml-rajapinta. Versio 0.5, 5.10. 2003, STM: Kansallinen terveyshanke, HL7 Finland ja VTT: Open CDA –hanke, Stakes: koodistopalvelin hanke.
- Lehtonen J. Valtakunnalliset palvelut sähköisen potilaskertomuksen tukena - Koodistopalvelin. Esitys PlugIT-seminaarissa, Stakes / luokituskeskus, 27.10.2003.
- MDF 1999. Beeler GW, Huff S, Rishel W, Shakir A-M, Walker M, Mead C, Schadow G: HL7 version 3 Message Development Framework, version 3.3, December 1999.
- Mykkänen J, Sormunen M, Karvinen K, Tikkanen T, Päiväniemi S. Component and Service Technology Families. Studies and Reports of the PlugIT project 1. Kuopio, 2004a.
- Mykkänen J, Häyrynen K, Savolainen S, Porrasmaa J. Standardien arviointi ja valinta terveydenhuollon sovellusintegraatioissa. PlugIT-hankkeen selvityksiä ja raportteja 3. Kuopio 2004b.
- Mykkänen J, Porrasmaa J, Rannanheimo J, Tikkanen T, Sormunen S, Korpela M, Häyrynen K, Eerola A, Häkkinen H, Toivanen M. Terveydenhuollon sovellusintegraatoratkaisujen määrittely. PlugIT-hankkeen selvityksiä ja raportteja 4. Kuopio, 2004c.
- Mykkänen J, Toroi T, Karhunen H, Virkanen H, Mäki H, Sormunen M, Rannanheimo J, Tuomainen M. Avointen integrointiratkaisujen hyödyntäminen, toteuttaminen ja testaus. PlugIT-hankkeen selvityksiä ja raportteja 5. Kuopio 2004d.
- OMG 2000. OMG Healthcare Domain Task Force: The CORBAmed Roadmap, OMG Document Number CORBAmed/2000-05-01, Version 2.0 (draft), 2000.
- PlugIT 2002. Integrointivaatimusten määrittely. Käyttäjä- ja käyttöoikeustiedot potilashallinnon ydinjärjestelmästä erillisjärjestelmään, käyttäjä- ja oikeuskomponenttirajapinta, luonnos 18.7.2002. PlugIT.
- PlugIT 2003e. PlugIT:n ydinpalvelu- ja työpöytäintegraatorajapintojen tekninen liittymämäärittely http- ja XML-tekniikoilla. Luonnos 21.10.2003, PlugIT.
- PlugIT 2004. PlugIT-hankkeen kotisivut. PlugIT-hanke, 2004. <http://www.plugin.fi/>.
- Porali M, Riekkinen A, Pohjolainen P, Mykkänen J, Toroi T, Kärkkäinen T-L, Eerola A. Ohjelmistotuotannon nykytilaselvitys 2003 - kohderyhmänä terveydenhuollon ohjelmistoyritykset ja organisaatiot. PlugIT-hankkeen selvityksiä ja raportteja 14. Kuopio, 2004.
- Ruotsalainen P, Ensio A. Sähköisen asiakas- ja potilasasiakirjojen säilytyksen ja kiistämättömyyden hyvä käytäntö. Osaavien keskustien verkon julkaisuja, 2003.
- Savolainen S. Terveydenhuollon ohjelmistojen yhteisten palvelujen standardit. Kuopion yliopiston selvityksiä H. Informaatioteknologia ja kauppatieteet 1. Kuopion yliopisto, 2004.
- Solbrig H (toim.). HL7 Common Terminology Services. HL7 Version 3 Standard, Health Level Seven, Inc., 2003.

- Sormunen M, Porrasmaa J, Rannanheimo J, Mykkänen J, Savolainen S. Terveystenhuollon avoimet sovellusrajapinnat - yhteiset perusratkaisut. PlugIT-hankkeen selvityksiä ja raportteja 7. Kuopio, 2004a.
- Sormunen M, Porrasmaa J, Silvennoinen R, Mykkänen J, Savolainen S, Rannanheimo J. Terveystenhuollon avoimet sovellusrajapinnat - käyttäjä- ja käyttöoikeusrajapinnat. PlugIT-hankkeen selvityksiä ja raportteja 9. Kuopio, 2004b.
- Stakes 2002. Sosiaali- ja terveydenhuollon käsitteitä tietojärjestelmien suunnittelua varten. Stakes, 2002. <http://www.stakes.fi/oske/terminologia/sanastot/kasite.htm>.
- TQS 2000. Lexicon Query Service Specification, Version 1.0. OMG Healthcare Domain Task Force, 2000.
- van der Linden H, Tange H, Talmon J, Hasman A. PropeR revisited. In: Baud R, Fieschi M, Le Beux P, Ruch P, eds. The new navigators: from Professionals to Patients. Proceedings of MIE2003. Amsterdam: IOS, 2003, p. 346-351.

LIITE 1.STAKESIN KOODISTOPALVELUN JA HL7 MDF:N TIETOELEMENTIT

Tietosisällössä on viitattu eri lähteisiin seuraavasti:

MDF = HL7 Message Development Framework, minimitaso koodistoratkaisuille (MDF 1999)

Koodistopalvelu = Stakesin koodistopalvelimen määrittelydokumentit (Koodistopalvelu 2003)

Sekä koodisto- että koodiarvo-elementteihin liitettävät ominaisuudet (Koodistopalvelu 2003):

- id:ST – tunnus (koodistolla OID, koodilla koodiarvo) (MDF-koodiston tunniste OID, MDF-koodiarvo?)
- type:ST – tyyppi (ks. alla)
- dataType:ST – tietotyyppi (esim ST, TS jne.)
- language:ST – kieli
- createDate, beginDate, expirtionDate, lastModifedDate:TS – luontipvm, voimassaolon alku, voimassaolon päättymisen, muutospvm
- lastModifedBy:ST - muuttaja

Atribuutin tyyppien nimet (Koodistopalvelu 2003), kukin koodisto tai koodiarvo sisältää useita attribuutteja, ja varsinainen arvo on elementin (attribute) sisältönä:

- status: tila (status)
- shortname: lyhyt selite (MDF: koodiston nimi)
- longname: pitkä selite
- abbreviation: lyhenne (MDF-koodiarvo)
- version: versio (MDF: koodiston versio)
- codetype: koodiston tyyppi (nimikkeistö, koodisto, luokitus, organisaatio)
- descriptor: kuvailu (MDF: koodiston kuvaus)
- parentID: yläluokka
- hierarchyLevel: hierarkiataso 0, 1, 2, 3, 4 (0=ylin)
- gategory: (voidaanko tasolle syöttää)
- leafnode: onko kyseessä lehtitaso
- child: alaluokka
- sibling: vastaavuudet
- source: lähde
- externallink: ristiviittaukset eri koodistoihin (CV)
- annotation: huomautus
- concept: käsite
- synonym: synonyymi
- definition: määritelmä
- semantic: semanttinen
- comment: kommentti
- instruction: käyttöohje
- inclusion: mukaan lukien

- exclusion: poislukien
- preferred: ensisijainen vastine
- footnote: alaviite
- index: hakusanat (mittayksikkö, erikoisala jne.)
- capitalvariant: kirjainkokovariantti
- wordordervariant: sanajärjestysvariantti
- singularvariant: yksikkövariantti
- pluralvariant: monikkovariantti
- difference: muu ero
- break: katkaisu
- source: merkijonon alkuperä
- historynote: historiallinen huomautus
- onlinenote: online huomautus
- publicmeshnote: yleinen MeSH huomautus
- consider: harkitse myös
- authorizer: laatija
- originator: perustaja
- maintainer: ylläpitäjä
- local: paikallinen

Lisäksi koodistosta tarvitaan tietoa koodiston koodien lukumäärästä.

MDF määrittelee koodiarvon vaihtoehtoiseksi merkkijonomuotoisen korvaustekstin kanssa.

Koodistokohtaisia tietosisällön erityispiirteitä (Koodistopalvelu 2003):

Organisaatioiden yksilöintiin toimipaikkarekisterissä käytetään organisaatitunnuksia, jotka ovat OID-muotoisia, ja rekisteri sisältää lisäattribuutteja:

- costCenter:ST kustannuspaikka
- officePostAddress:ST toimipaikan postiosoite
- officeTelephone:ST toimipaikan puhelin
- officeFax:ST faksi
- officeMail:ST toimipaikan sähköposti
- officeMunicipality:CV toimipaikan kunta
- CV toimipaikan erikoisala
- CV toimipaikan tarjoamat palvelut
- ST toimipaikan läheteiden toimitus sähköpostilla
- ST toimipaikan läheteiden toimitus postilla
- ST toimipaikan käyntiosoite
- ST toimipaikan rakennus
- ST toimipaikan kerros
- ST toimipaikan laskutusosoite
- ST toimipaikan info sivut webbi
- ST toimipaikan ajanvaraus

LIITE 2. TEKNISEN MÄÄRITYKSEN OPERAATIOLISTAUS (V2.0)

Kunkin operaation perässä taso, jolla operaatio on mukana. Lisäparametrit, jotka eri operaatioissa hyväksytään, riippuvat siitä, mitä tasoja palvelu toteuttaa. Minimitason vaaditut operaatio on alle-
viivattu.

CodeService-rajapinta

GetSupportedCodeSystems: perustaso

GetSupportedServices: perustaso

GetInfo: perustaso

Codeset-rajapinta

LookupCodesByDesignation: minimitaso

ListCodes: minimitaso

LookupCodes: perustaso

GetSupportedCodesetServices: perustaso

IsCodeValid: perustaso

GetCodesetInfo: perustaso

ListLanguages: monikieliset koodistot

GetHierarchyDepth: hierarkkisten koodistojen syvyyden kysely

GetCodes: vapaa tietosisältö

GetSupportedAttributes: vapaa tietosisältö

Code-rajapinta

GetDesignation: minimitaso

GetParent: hierarkkisten koodistojen käsittely

GetHierarchyLevel: hierarkkisten koodistojen käsittely

GetStatus: koodien paikallisuus ja tila

GetLocal: koodien paikallisuus ja tila

LookupCompleteCodedConcept: vapaa tietosisältö

LookupProperties: vapaa tietosisältö

LIITE 3. MÄÄRITYKSEN TASOJEN VAATIMUKSET TOTEUTUKSILLE

Eri tasojen toteutusten on käytävä ilmi perustason operaatioista `CodeService:GetSupportedServices` ja `Codeset:GetSupportedCodesetServices`.

Minimitaso:

- `Codeset:LookupCodesByDesignation`: hyväksyttävä `matchtext`-parametri, palautettava tarkasti `matchTextiä` vastaavien (`partial=0` oletus, ei tarvitse antaa) koodien tunnisteet ja selitteet oletuskielellä
- `Codeset:ListCodes`: hyväksyttävä `howMany` ja `From`-parametrit, palautettava `more`-tieto sekä koodien tunnisteet ja selitteet oletuskielellä
- `Code: GetDesignation`: palautettava koodiarvoa vastaava selite oletuskielellä

Perustaso (base):

(minimitason lisäksi)

- `CodeService:GetSupportedCodeSystems`: palautettava tuetut koodistot
- `CodeService:GetSupportedServices`: palautettava palvelussa tuetut palvelutasot
- `CodeService: GetInfo`: palautettava tietoa palvelusta (voi palauttaa myös tyhjän)
- `Codeset:LookupCodesByDesignation`: minimitason lisäksi hyväksyttävä `partial=1` -parametri, palautettava myös osittain (alusta) `matchTextiä` vastaavat, palautettava tulokset `sortBy`-parametrin (arvona `id` tai `shortname`) mukaisen elementin mukaan lajiteltuna
- `Codeset:LookupCodes`: hyväksyttävä `matchtext` ja `howMany`-parametrit, oletuksena haku tapahtuu osittaisesta koodiarvosta (`partial=1`) (perustasolla ei muita hakuarvoja), palautettava tulokset `sortBy`-parametrin (arvona `id` tai `shortname`) mukaisen elementin mukaan lajiteltuna
- `Codeset:ListCodes`: palautettava tulokset `sortBy`-parametrin (arvona `id` tai `shortname`) mukaisen elementin mukaan lajiteltuna
- `Codeset:IsCodeValid`: toteutettava operaatio
- `Codeset:GetSupportedCodesetServices`: palautettava palvelun koodistossa tukemat palvelutasot
- `Codeset:GetCodesetInfo`: palautettava tietoa koodistosta (voi palauttaa myös tyhjän)

Monikieliset koodistot (multilingual):

(perustason lisäksi)

- `Codeset:LookupCodesByDesignation`: hyväksyttävä `language`-parametri, palautettava selite halutulla kielellä
- `Codeset:ListCodes`: ks. edell.
- `Codeset:LookupCodes`: ks. edell.
- `Code:Designation`: ks. edell.
- `Codeset:ListLanguages`: toteutettava operaatio

Hierarkkisten koodistojen käsittely (hierarchy):

(perustason lisäksi)

- `Codeset:LookupCodesByDesignation`: hyväksyttävä `parentId`-parametri, palautettava vain annettua juurta seuraavalla tasolla olevat tulokset
- `Codeset:ListCodes`: ks. edell.
- `Codeset:LookupCodes`: ks. edell.
- `Codeset:GetHierarchyDepth`: toteutettava operaatio
- `Code:GetParent`: toteutettava operaatio
- `Code: GetHierarchyLevel`: toteutettava operaatio

Paikalliset koodit ja koodien tila (status):

(perustason lisäksi)

- Codeset:LookupCodesByDesignation: hyväksyttävä vähintään status="1" (active) rajausta tehtävissä haissa, voi hyväksyä myös local-rajaukset
- Codeset:ListCodes: ks. edell.
- Codeset:LookupCodes: ks. edell.
- Code:GetStatus: toteutettava operaatio
- Code:GetLocal: toteutettava operaatio

Vapaa tietosisältö (freeElements):

(perustason lisäksi)

- Codeset:LookupCodesByDesignation: hyväksyttävä propertyCodeList-parametri, palautettava pyydetty sisältöelementit, myös halutulla kielellä jos lisäksi toteutetaan multilingual-taso ja pyydetty elementti tukee useita kieliä
- Codeset:ListCodes: ks. edell.
- Codeset:LookupCodes: ks. edell.
- Codeset:GetCodes: toteutettava operaatio
- Codeset:GetSupportedAttributes: toteutettava operaatio
- Code:LookupCompleteCodedConcept: toteutettava operaatio
- Code:LookupProperties: toteutettava operaatio

Edistyneet haut (advSearch):

(perustason lisäksi)

- Codeset:LookupCodesByDesignation: hyväksyttävä kaikki partial-arvot, hyväksyttävä myös useita search-parametreja, jos myös freeElements on toteutettu, sortBy-parametrina lajittelussa voidaan käyttää myös freeElements-tason elementtien nimiä
- Codeset:LookupCodes: ks. edell.
- Codeset:ListCodes: jos myös freeElements on toteutettu, sortBy-parametrina lajittelussa voidaan käyttää myös freeElements-tason elementtien nimiä

LIITE 4. TEKNISEN MÄÄRITYKSEN XML-ELEMENTIT

Minimitasolla tarvittavat elementit ja attribuutit on lihavoitu. Elementit ovat päätasolla, niiden attribuutit sisennettyinä. Elementtejä on 26.

attribute - vapaassa tietosisällössä Koodistopalvelun XML-siirtosanomat-määrittelyn mukaisesti yhden koodin jonkin tietoelementin sisältävä elementti, voi sisältää alla lueteltujen lisäksi `dataType`, `createDate`, `beginDate`, `expirationDate`, `lastModifiedDate`, `lastModifiedBy` -attribuutit ja lisäattributteja (esim. `attribute1`, `attribute2`), joiden merkitys on tarkennettava sisältömäärittelyssä tai toteutuskohtaisesti. Tässä määrittelyssä käytetyt attribuutit:

- `type` (pakollinen): attribuutin nimi, esim. "shortname", "longname"
- `language`: attribuutin kielikoodi, ks. `language` -> id.
Vapaassa tietosisällössä `language` voi liittyä ainakin `abbreviation`, `shortname` ja `longname` -tyyppisiin attribuutteihin

current - status-tasolla sisältönä tarkka päivämäärä, jona voimassaolevia koodeja halutaan palautettavaksi haussa tai listauksessa. Päivämäärän esitysmuoto on YYYY-MM-DD, esim. 2004-04-12. Koodin voimassaolon alkamis- ja päättymispäivä lasketaan kuuluvaksi voimassaoloaikaan. Päivämäärämuoto on kuvattu tarkasti standardissa ISO8601:2000(E): Data elements and interchange formats - Information interchange - Representation of dates and times.

description - sisältönä palvelun tai koodiston kuvausteksti

display - jos operaatio palauttaa vapaata tietosisältöä, palautettavien (näytettävien) ominaisuuksien lista

find - yhdessä haussa haun suorittamiseen tarvittavat tiedot sisältävä elementti

from - jatkohaussa tai -listauksessa ensimmäisenä palautettavan koodin tunniste

howMany - sisältönä listauksissa ja hauissa palautettavien tulosten maksimimäärä

interface - sisältönä kutsuttavan rajapinnan nimi

language - sisältönä kielen näytettävä nimi, käytetään kielijoukkoja käsittelevissä operaatioissa, yleensä kieli ilmoitetaan toisen elementin `language`-attribuutissa

- `id`: (pakollinen) kielen tunniste (SFS-ISO-639, Alpha-2 code)

local - sisältönä 0 (ei-paikallinen koodi) tai 1 (paikallinen koodi), kun tehdään hakua jonka on palautettava vain paikallisia tai ei-paikallisia koodeja. Local-elementin puuttuessa paikallisuutta ei huomioida (palautetaan sekä paikalliset että ei-paikalliset).

matchtext - sisältönä haussa käytettävä hakuarvo

- `language`: jos hakuarvo kohdistuu tiettyyn kieleen

- `partial`: hakutapa: 0-tarkka osuma, 1-osuma alusta, 2-osuma missä tahansa alimerkkijonossa

method - sisältönä kutsuttavan operaation nimi

param - kutsussa tarvittavat parametrit sisältävä elementti

parentId - sisältönä haussa käytettävän hierarkian juurielementin koodiarvo

property - sisältönä tietosisällön yhden elementin nimi

language: jos sama elementti saatavana usealla kielellä, kielen tarkenne. Ainakin abbreviation, shortname ja longname-tyyppisillä attribuuteilla voi olla useita tuettuja kieliä.

propertyCodeList - vapaassa tietosisällössä halutun tietosisällön elementtien nimet (property) sisältävä ominaisuus

request - palvelukutsu, sisältää kaikki palvelun tarvitsemat parametrit

response - vastaus palvelukutsuun, sisältää kutsun paluuarvot tai virhetiedot

server - sisältönä koodistotarjoajan (tuotteen) nimi
version: tuotteen versio (vapaaehtoinen)

service - yhden tuetun palvelutason tiedot, sisältönä selväkielinen nimi

id: attribuutin arvona "base", "multilingual", "freeElements", "advSearch", "hierarchy" tai "status"

version: attribuutin arvona vastaavan liittymämäärittelydokumentin versio tai tunnistus

status - sisältönä koodin tila, jota vastaavat tulokset palautetaan haussa, 1-active, 0-work, 2-delete. status-elementin puuttuessa tilaa ei huomioida (palautetaan sekä aktiiviset, ehdotetut että poistettut koodit). "Poistettu" koodi tarkoittaa, että koodia ei enää tulisi käyttää myöskään aikaisemmin tehdyissä viittauksissa koodistoon.

sortBy - sisältönä hakutulosten palauttamisen järjestyksen kertova attribuutti (lajittelukenttä). Perustasolla attribuutin arvona voi olla id tai shortname. Oletusarvo (parametrin puuttuessa tai minimitasolla) id. Jos toteutetaan sekä advSearch että freeElements-tasot, parametrin arvona voi olla minkä tahansa tuetun sisältöelementin nimi (sisältömäärittelyksen mukaisesti). Toteutuskohtaisesti dokumentoidut poikkeamat voidaan sallia tilanteessa, jossa on toteutettu sekä advSearch että freeElements eikä lajittelu jokaisen kentän mukaan ole järkevää (esim. suuri määrä samoja arvoja).

term - yksi koodiarvo, sisältönä näytettävä arvo (selite, shortname)

id: (pakollinen, koodiarvo, esim. "A01.0")

language: selitteen kieli

termItemEntry - yksi koodiarvo vapaalla tietosisällöllä, sisältää Open CDA määrittelyksen mukaisesti yhden tai useampia attribute-elementtejä, joissa eri ominaisuuksien arvot

id: pakollinen, koodiarvo, esim. "A01.0")

termSystem - yhden tuetun koodiston tiedot, sisältönä (esim. paluuarvoissa) näytettävä nimi (vast. OpenCDA longname)

id: (pakollinen, esim. oid)

language: kielen koodi jos vain yksi kieli määritelty koodistossa tai operaatio kohdistuu vain tiettyyn kieleen

version: koodiston versio (jos ei käy ilmi id:stä)

value - yksinkertaisen operaation esim. numeerinen paluuarvo

LIITE 5. KOMMENTOINTIKIERROKSELLA SAADUT KOMMENTIT JA KYSYMYKSET VASTAUKSINEEN

Tämä liite sisältää HL7 Finland kommentointikierroksella 3.12.2004 – 7.1.2005 saadut kommentit ja tarkentavat kysymykset ja niiden vastaukset, jotka käsiteltiin Common Services SIG kokouksessa 10.1.2005.

Miksi eri tasoilla käytetään saman asian (koodi) kuvaamiseen term ja termItemEntry-elementtejä?

- term-elementti on peräisin minimitason vaatimuksesta, ja sisältää minimitietosisällön. Vapaan tietosisällön tasolla on käytettävissä sama laajempi elementti kuin koodistopalvelun siirtosanomissa (termItemEntry). Kahden elementin käyttö monimutkaistaa toteutusta hie- man, mutta on järkevää määritellä eri elementit, koska niillä on selkeästi eri käyttötarkoitus ja -tapa.

Onko hakuarvon (matchtext-elementti) kirjainkoolla merkitystä hauissa?

- LookupCodesByDesignation ja LookupCodes –operaatioiden kuvaus-kohdassa on mainittu, että ”hakuarvojen pienet ja isot kirjaimet vastaavat toisiaan”. Malli on otettu TQS-määrittämisestä, eikä ole nähty tarvetta rajata hakuja kirjainkoon perusteella. Jos tällainen tar- ve ilmenee, voidaan määritellä uusi hakuattribuutti.

Jos hakuparametrina annetaan parentId (hierarkkiset koodistot), haetaanko seuraavalta hierarkiata- solta vai kaikilta alemmilta?

- Esim. liitteessä 3 on mainittu: ”palautettava vain annettua juurta seuraavalla tasolla olevat tulokset”. Erillisen syvyyshakuparametrin lisääminen määrittämisessä on mahdollista, jos sitä tarvitaan, mutta syvän haun voi toteuttaa myös rekursiivisesti (vaatii useita kutsuja). Hie- rarkkisen haun sijaan voidaan myös tehdä haku hierarkiata-son sisältävään tietokenttään, jos toteutetaan ”advSearch” ja ”freeElements”-tasot, ja ko. koodisto tukee hierarkiata-son sisäl- tävää tietokenttää.

Miten koodeille voi lisätä itse määriteltäviä ominaisuuksia esim. oman sovelluksen käyttöön?

- Sovelluskohtaisia ominaisuuksien laajennuksia ei ole käsitelty sisältömäärittämisessä. Suosi- tuksena on sama nimeämiskäytäntö kuin ProfileAccess-rajapinnassa: <proper- ty>[yritys.fi]ohjelmistoversio</property>. Sovelluskohtaiset laajennukset sisältömäärittä- miin on dokumentoitava.

Jos ominaisuuksien tunnisteetkin olisivat omista koodistoistaan, olisi mahdollista määritellä omi- naisuuskoodistojen välisiä vastaavuuksia ja saada sovellukset ymmärtämään paremmin toisiaan.

- Kunkin koodiston sisältömäärittäminen on myös ”tunnisteluettelo”. Tunnisteiden lähteenä on käy- tetty Koodistopalvelun siirtosanomien tietoelementtien nimiä (Liite 1). Kussakin sisältömää- rityksessä on kuvattava, millä nimellä mitään ominaisuutta kutsutaan, joten myös muuta voisi käyttää.

Joskus voisi olla hyvä tietää millä tasolla koodi on paikallinen, esim. toiseen organisaatioon lähetet- täessä tuntee vastaanottaja koodin; voisiko käyttää koodien originator-attribuuttia (vai authorizer) – luonut organisaatio?

- Liitteessä 1 kuvatuista elementeistä: lastModifiedBy, source(?), authorizer, originator, main- tainer olisivat mahdollisia lähteen yksilöinnissä. Mutta päättely (tuntee vastaanottaja koo- din) näistä ominaisuuksista vaatii joka tapauksessa paikallista sovelluslogiikkaa. Local- elementtiä ei toistaiseksi käytetä kansallisessa koodistopalvelussa. Asiaan liittyy lisäksi se,

onko paikallisia koodeja lisättäessä koko koodisto tunnistettava tämän jälkeen eri tavalla kuin alkuperäinen koodisto. Mikäli näin on ja paikallisia koodeja halutaan käyttää, suositellaan, että:

- tässä rajapinnassa paikallisia koodeja sisältävän koodiston termsystem-elementin id-attribuutissa käytetään koodiston yleistä tunnistetta,
- termsystem-elementin version-attribuutissa käytetään koodiston paikallista tunnistetta,
- jos paikallinen koodi (saadaan selville esim. isLocal-operaatiolla) siirretään toiseen järjestelmään, sen koodiston yksilöintiin käytetään vain koodiston paikallista tunnistetta.

Liitteessä 1 on kirjoitusvirheitä

- Kirjoitusvirheet ovat peräisin kansallisen koodistopalvelun XML-sanomien määrittelystä (1.2.246.777.11.2004.6), ja virheet on korjattu tämän dokumentin muihin osiin sekä tietävästi lopulliseen siirtosanomien määrittelyyn. Liite 1 on informatiivinen, ja attribuuttien nimissä tulee käyttää virheettömiä muotoja.