

Perusterveydenhuollon avohoidon potilasryhmitys (APR)

Rajapintamääritys

SerAPI-projekti	
Yhteyshenkilö	Heli Luostarinen (Heli.Luostarinen@uku.fi)
Dokumentin versio	1.0
Päiväys	2.6.2006

Sisällysluettelo

1	Johdanto ja määrittelyn tavoitteet	4
2	Käsitteet ja taustaa.....	5
2.1	Keskeiset käsitteet	5
2.2	APR	6
2.2.1	APR-ryhmittelyn tavoitteet ja hyödyt	6
2.2.2	Kontakti, hoitopisodi ja hoitoketju APR-ryhmittelyssä	7
2.2.3	APR:n luokitukset.....	9
3	Rajapinnan käytön edellytykset	10
3.1	Oikeuksiin, tietoihin ja toimintatapoihin liittyvät vaatimukset.....	10
3.2	Tekniikkaan liittyvät vaatimukset	10
4	Rajapintaratkaisun kuvaus	12
4.1	Sovellusosapuolet ja yhteistoiminnallisuuden arkkitehtuuri	12
4.2	Tekninen arkkitehtuuri.....	13
5	Toiminnallisuus.....	14
5.1	Roolit ja vastuut.....	14
5.2	Palvelun käyttötavat	14
6	Rajapinta ja sen tietosisältö	16
6.1	Minimitason tietosisältö ja laajennukset.....	17
6.1.1	Minimitaso.....	17
6.1.2	Laajennukset.....	17
6.2	Kutsu (Get_pAPR)	17
6.2.1	Kontaktin syy (contactReason).....	18
6.2.2	Toiminto (activity)	18
6.2.3	Kontaktityyppi (contactType).....	19
6.2.4	Kontaktin luonne (contactCharacteristic).....	19
6.2.5	Potilaan ikä (age)	20
6.2.6	Potilaan sukupuoli (gender).....	20
6.2.7	Kutsun laajennus (inputExtension)	20
6.3	Vastaus (Get_pAPRResponse).....	21
6.3.1	pAPR-ryhmittelyn tulos (pAPRResult).....	21
6.3.2	pAPR-ryhmä (pAPR)	22
6.3.3	pAPR-ryhmän painokerroin (weight)	22
6.3.4	Tilaliput (statusFlags).....	23
6.3.5	Vastauksen laajennus (outputExtension)	25
6.4	Virhetilanteet.....	25
6.5	Esimerkkejä SOAP-viesteistä	26
7	Toteutuskohtaiset asiat	28
7.1	Rajapinnan toteutuksessa huomioitavat asiat.....	28
7.2	Turvallisuusnäkökohdat.....	28
7.3	Määrittelyn mukaisuus	29
8	Määrittelyn jatkokehitys ja toteutukset	30

Liite 1: WSDL-rajapintakuvaus

Versiohistoria

Versio:	Pvm:	Laatijat:	Selitys:
Versio 1.0	2.6.2006	Heli Luostarinen Esa Paakkanen Assi Pöyhölä Juha Mykkänen	Julkistettava versio

Esipuhe

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

Kiitokset kaikille tähän työhön osallistuneille, erityisesti Timo Pessille Datawell Oy:stä ja Outi Elonheimolle DRG Medical Systems Oy:stä.

1 Johdanto ja määrittelyn tavoitteet

Tämä dokumentti on avoin rajapintamäärittely perusterveydenhuollon Avohoidon potilasryhmityksen (APR) liittämiseksi potilastietojärjestelmiin. Dokumentti sisältää APR-ryhmittelyn tarjoavan sovelluspalvelun liittymämäärittelyn WSDL-pohjaisena ratkaisuna. WSDL-rajapintakuvaus on liitteessä 1.

APR on luokitusjärjestelmä, jonka avulla kuvataan terveyskeskusten potilasryhmät ja sairauksien hoitamiseksi tehdyt toimenpiteet. (DMS 2006b). Tarkemmin APR-ryhmittely on kuvattu luvussa 2.

Dokumentin sisältö painottuu tekniseen liittymämäärittelyyn ja siten se on tarkoitettu ensisijaisesti rajapintojen toteuttajan ja hyödyntäjän referenssiksi ohjelmointityötä varten. Varsinainen liittymämäärittely sisältää rajapinnan toiminnallisuuden ja tietosisällön kuvaukset (luvut 5 ja 6). Dokumentti sisältää minimitason määrittelyn, jolla ryhmittely saadaan käyttöön. Sen lisäksi mukana on eipakollisten, mutta tunnistettujen ominaisuuksien määrittelyt sekä varautumisen mahdollisia laajennuksia varten. Dokumentissa myös kuvataan asiat, joihin tulisi ottaa kantaa toteutuskohtaisesti (luku 7). Integroitumisen lisäksi teknistä dokumentaatiota voidaan käyttää pohjana toteutuskohtaista dokumentaatiota laadittaessa tai tietojärjestelmien tarjouspyynnön liitteenä.

Ei-teknisistä piirteistä dokumentissa kuvataan rajapinnan käytön edellytykset eli vaatimukset, jotka on otettava huomioon ennen toteutusta (luku 3), sekä ratkaisun pääperiaatteet arkkitehtuuritasolla (luku 4). Lopussa esitetään ajatuksia määrittelyn jatkokehityksestä.

Rajapintamäärittelyn tavoitteena on helpottaa APR-ryhmittelyn liittämistä osaksi perusjärjestelmiä. Toiminnan kannalta tavoitteena on auttaa tuotteistuksen kautta arvioimaan väestön palveluiden tarvetta, keskittämään resursseja todellisen tarpeen mukaan sekä ohjaamaan tarvittavia toimintakäytännömuutoksia ja seuraamaan niitä.

2 Käsitteet ja taustaa

2.1 Keskeiset käsitteet

APR, Avohoidon potilasryhmitys

Perusterveydenhuollon avohoidon potilasryhmitys. APR tarjoaa terveyskeskuksen operatiiviselle johdolle monenlaista luokiteltua ja vertailukelpoista toimintatietoa potilasryhmistä ja niiden hoitokäytännöistä. (DMS 2006a) (vrt. DRG-luokitus)

APR-ryhmä (episodiAPR-ryhmä)

APR-ryhmä on terveyskeskuksen määritelty tuote. Se kuvaa eri ammattiryhmien aikaansaamaa hoitoprosessia, joka kohdistuu potilaan tiettyyn sairauteen tai tiettyihin sairauksiin. (DMS 2006a)

ASL, Avohoidon sairausluokitus

ASL ilmaisee käynnin syyn. Sairaudet on jaettu 42 perusterveydenhuollon potilasainekselle sovellettuun elinryhmäkohtaiseen ryhmään. ASL-luokka voidaan myös johtaa ICPC- ja ICD-10 -diagnoosiluokituksista.

ATL, Avohoidon toimintoluokitus

ATL ilmaisee suoritettavat tutkimus- ja hoitotoimenpiteet (toiminnot). Kullakin työyksiköllä on omat toimintoluokituksensa, jotka sisältävät sekä kontaktin aikana tehtävät toimenpiteet että jatkotoimenpiteet. ATL-luokka voidaan johtaa myös NCSP-toimenpideluokituksista, niiltä osin, kuin NCSP kuvaa perusterveydenhuollon toimenpiteitä.

DRG (Diagnosis Related Groups)

Erikoissairaanhoidon potilasluokittelujärjestelmä. DRG:n avulla pystytään tuotteistamaan ja hinnoittelemaan erikoissairaanhoidon palvelut sekä seuraamaan erikoissairaanhoidon kustannuksia ja laatua. (Kuntaliitto 2006a)

ICD-10 (10th revision of the International Classification of Diseases)

Maaailman terveysjärjestön (World Health Organization, WHO) määrittämä tautiluokitus. Luokituksen suomalaista versiota käytetään Sosiaali- ja terveysministeriön määräyksen (STM määräyskokoelma 1995:81) mukaisesti tautien ja kuolinsyiden merkitsemisessä potilasta koskeviin asiakirjoihin Suomessa. Tautiluokitusta käytetään siten kliinisessä työssä, tilastoissa, sosiaalivakuutuksessa sekä lisäksi tutkimuksessa. ICD-10 tekijänoikeudet ovat WHO:lla ja Stakesilla. (Stakes 2005)

ICPC (International Classification of Primary Care)

Käytetään perusterveydenhuollossa tapahtuneiden vastaanottokäyntien syiden (hoidon tarve), diagnoosien ja hoitoprosessin luokittelussa. ICPC:lla voidaan luokitella potilaan ilmaisema käynnin syy tai hoidon tarve, hoidon antajan arvio tai diagnoosi potilaan terveysongelmasta ja hoitoprosessin aikana tapahtuneet diagnostiset ja terapeuttiset (hoidolliset) toimenpiteet. (PlugIT 2004a) Käytännössä tätä käytetään ainoastaan diagnoosiluokituksena, ei toiminnan kuvaamiseen.

NCSP (NOMESCO Classification of Surgical Procedures)

Pohjoismainen leikkausluokituskoodisto, jonka alun perin on määrittänyt ja julkaissut The Nordic Medico-Statistical Committee (NOMESCO) vuonna 1996. Suomalainen versio (NCSP-F) on otettu käyttöön vuonna 1997. (NOMESCO 2004)

pAPR-ryhmä (perusAPR-ryhmä)

Kontaktikohtaisen ryhmittelyn tuloksena saatava välituote. pAPR-ryhmiä muodostetaan yksi tai useampi jokaisesta käynnistä. Sairauteen liitetään käynnin aikana sen hoitamiseksi tehdyt toimenpiteet. pAPR-ryhmiä muodostuu niin monta kuin käynnin aikana on käsitelty eri sairauksia.

Tuotteistaminen

Terveydenhuollon ammattilaisten tietyn sairauden tai tiettyjen sairauksien hoitamiseksi tekemien potilastyöhön liittyvien tehtävien kuvaaminen ja yhteenliittäminen (DMS 2006d).

2.2 APR

Avohoidon potilasryhmitys (APR) on DRG Medical Systems Oy:n (DMS) yhdessä usean terveyskeskuksen kanssa kehittämä luokitusjärjestelmä, jonka avulla kuvataan terveyskeskusten potilasryhmät ja sairauksien hoitamiseksi tehdyt toimenpiteet. (DMS 2006b)

2.2.1 APR-ryhmittelyn tavoitteet ja hyödyt

Avohoidon potilasryhmitystä käytetään (DMS 2006d)

- potilasmateriaalin ja sen vaatavuuden kuvaamiseen vakioinnin avulla (case mix -tarkastelu)
- toimintakäytäntöjen kuvaamiseen
- tuottavuus- ja tehokkuuslaskelmiin
- väestön palvelujen tarpeen arviointiin
- alueellisten hoitoketjujen kuvaamiseen (perusterveydenhuolto + erikoissairaanhoito)

Avohoidon potilasryhmituksen hyödyt (DMS 2006d):

- Työntekijöiden, toimintayksiköiden, terveyskeskusten tuottavuus- ja toimintakäytäntövertailu - yksiköiden välillä (benchmarking)

- oman toiminnan seuranta aikasarjassa
- tuottavuuseroista voidaan porautua toimintakäytäntöeroihin
- Resurssien kohdentaminen ja tasaaminen sekä tarvittavien hoitokäytäntömuutosten havainnointi
- Toiminnan muutosten vaikutusten tarkastelu ja objektiivinen arviointi
 - tarkoitettu jatkuvaan terveystieteiden toiminnan seurantaan
- Tuotteistaminen luo perustan
 - kustannuslaskennalle
 - toiminnan vaikuttavuuden tarkastelulle potilasryhmäkohtaisesti
 - riskipotilaiden tunnistamiselle
- Potilaan hoitoketju voidaan kuvata läpi koko terveydenhuoltojärjestelmän linkittämällä APR DRG:hen

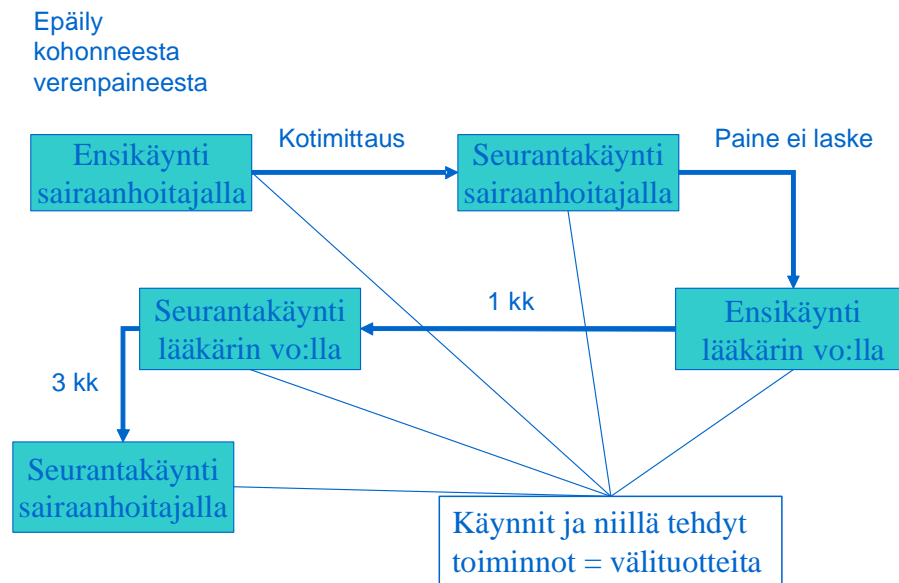
APR-luokituksen kirjaamisesta on välillistä hyötyä potilastietojärjestelmän käyttäjälle, koska ryhmittelyä käytetään toiminnan ja toimintakäytäntöjen kehittämiseen, johtamiseen ja seurantaan. Lisäksi APR-ryhmitys antaa mahdollisuuden kuvata esim. käyttäjän tai yksikön potilasmateriaalia ja sen vaatavuutta sekä kuvata näiden hoitamiseksi tehtyjä toimenpiteitä ja niiden määrää sekä yksiköiden toimintakäytäntöjä ja työnjakoa.

2.2.2 Kontakti, hoitoepisodi ja hoitoketju APR-ryhmittelyssä

Kontakti

Kontakti on hoitotapahtuma, joka on hoitoa antavan sosiaali- tai terveydenhuollon ammattihenkilön ja asiakkaan välinen yksittäinen vuorovaikutustilanne (Stakes 2006). Kontakti voi olla esimerkiksi vastaanottokäynti (ks. kuva 1), kotikäynti, ryhmäkäynti, puhelu tai sähköinen yhteydenotto.

Kustakin terveystieteiden avovastaanoton potilaskontaktista (esim. käynti, puhelu) kerätään potilas- ja toimintatiedot (eli sairauteen liitetään kontaktin aikana sen hoitamiseksi tehdyt toimenpiteet) ja suoritetaan ryhmittely. Tämän kontaktikohtaisen ryhmittelyn tuloksena saadaan pAPR-ryhmä (perusAPR-ryhmä, väli-tuote). Jos kontaktin aikana on käsitelty useampaa sairautta, kontaktikohtaisen ryhmittelyn tuloksena saadaan niin monta pAPR-ryhmää kuin kontaktin aikana on käsitelty eri sairauksia. (Pessi 2006)



Kuva 1. APR-tuote: verenpainepotilas (DMS 2006d)

Hoitoepisodi

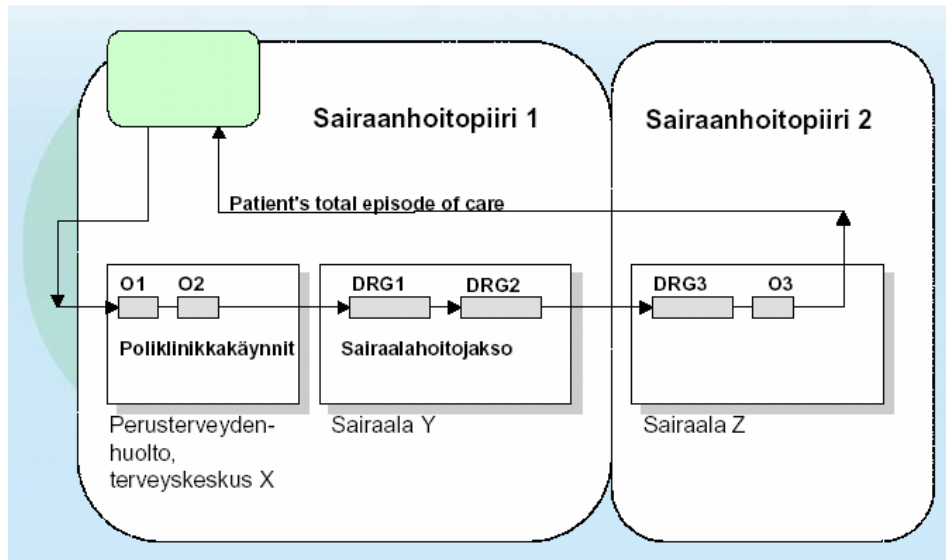
Hoitoepisodi on hoitoprosessi, joka on organisaatiokohtainen potilaan tautiin tai terveysongelmaan liittyvä ajanjakso. Hoitoepisodi alkaa, kun potilas ensi kertaa ilmaisee ongelman hoidonantajalle, ja päättyy viimeiseen kyseiseen tautiin tai terveysongelmaan liittyvään yhteydenottoon. (Stakes 2006)

Hoitoepisodikohtainen APR-ryhmä (episodiAPR, APR-tuote) muodostetaan potilaan kaikista toisiinsa samaan sairauteen, tai yhteenliittyviin sairauksiin, liittyvistä kontakteista (pAPR-ryhmistä, välituotteista). Potilaalla voi olla useita rinnakkaisia hoitoepisodeja. EpisodiAPR vaatii tietojen yhdistämistä mahdollisesti pitkältikin ajanjaksolta. (Pessi 2006) Kuvassa 1 on esimerkki yhdestä hoitoepisodista.

Hoitoketju

Hoitoketju on saman asiakkaan tiettyyn ongelmakokonaisuuteen kohdistuva, sosiaali- ja terveydenhuollon organisaatorajat ylittävä, suunnitelmallinen ja yksilöllisesti toteutuva hoitoprosessien kokonaisuus (Stakes 2006).

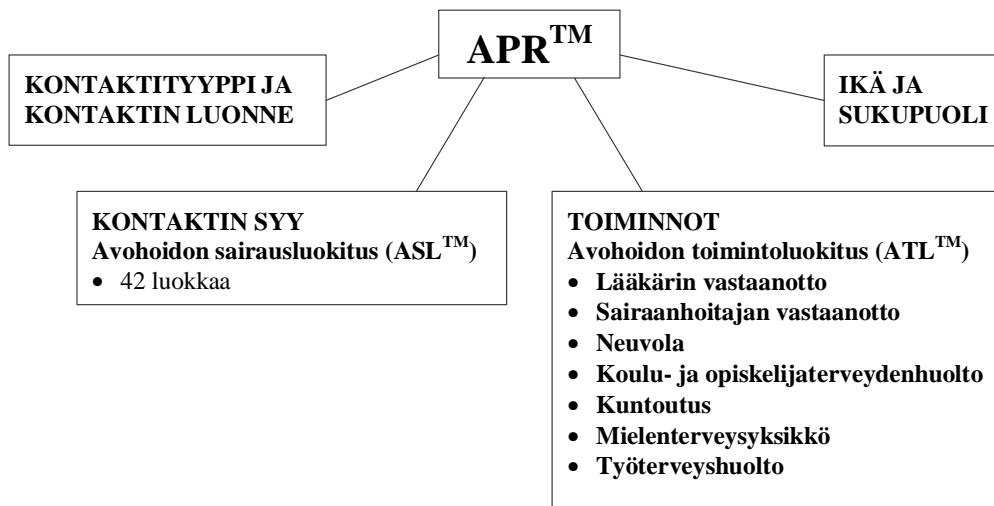
Potilaan saama tiettyyn kokonaisuuteen kuuluva hoito voidaan kuvata läpi koko terveydenhuoltojärjestelmän linkittämällä hoitoepisodikohtaiset APR-ryhmät erikoissairaanhoidon DRG (Diagnosis Related Groups) -ryhmiin (ks. kuva 2).



Kuva 2. Koko hoitoprosessin integrointi (DMS 2005)

2.2.3 APR:n luokitukset

APR käyttää neljää perusluokitusta: avohoidon sairausluokitus (ASL), avohoidon toimintoluokitus (ATL), kontaktityyppi ja kontaktin luonne. (DMS 2006b) Niiden tiedot kerätään kaikista terveyskeskuksen avovastaanottojen potilaskontakteista. Lisäksi potilaan ikä ja sukupuoli vaikuttavat potilasryhmän (APR-ryhmän) määräytymiseen (ks. kuva 3). (DMS 2006c)



Kuva 3. Avohoidon potilasryhmituksen perusluokitukset (mukailien DMS 2005)

Käytettävät perusluokitukset ovat saatavilla osoitteessa <http://www.apr.fi/apr.htm>. 23.3.2006 päivätty versio perusluokituksista löytyy myös SerAPI-hankkeen sivulta osoitteesta http://www.uku.fi/tike/his/serapi/mater/APR-perusluokitukset_230306.xls.

ASL voidaan johtaa ICPC- ja ICD-10 -diagnoosiluokituksista ja ATL NCSP-toimenpideluokituksista. ICPC, ICD-10 ja NCSP -luokitusten käyttämättömyys tekee APR:n käyttöönotosta helppompaa monille ko. luokituksia käyttäville terveyskeskuksille (DMS 2006d). Luokitusmuunnokset tekee APR-ryhmittelijä tai sen esikäsittelyvaihe (Pessi 2006).

3 Rajapinnan käytön edellytykset

3.1 Oikeuksiin, tietoihin ja toimintatapoihin liittyvät vaatimukset

Tähän lukuun on koottu vaatimukset, jotka liittyvät APR-ryhmittelypalvelun käyttöönottoon asiakassovelluksessa, kuten perusterveydenhuollon potilastietojärjestelmässä. Nämä seikat tulee huomioida ennen rajapinnan hyödyntämistä organisaatioissa ja ennen toteutustyötä tietojärjestelmissä.

APR-ryhmittelyn käyttö edellyttää käyttöoikeutta sekä APR-ryhmittelyn (luokittelujärjestelmä) että APR-ryhmittelijän (ohjelmisto) omistajilta. APR-ryhmittelyn omistaa tällä hetkellä DRG Medical Systems Oy, mutta omistusoikeus on luultavasti siirtymässä vielä julkistamattomalle kansalliselle toimijalle. Datawell Oy tarjoaa APR-ryhmittelijän (ohjelmiston) sisältävää tuotetta.

Palvelua hyödyntävässä organisaatiossa myös tulee kiinnittää huomiota tietojen kirjaamiseen. Ryhmittelyn onnistuminen edellyttää potilaan sairauksien ja toimenpiteiden huolellista kirjaamista ja koodaamista sairauskertomukseen, jotta ryhmittely voidaan luotettavasti suorittaa ja automatisoida. Luokittelun koodaamisessa käytetään ProWellness Oy:n diabetes-tietojärjestelmän yhteydessä toimivaa syöttöalustaa. APR-ryhmittelyyn tarvittavat tiedot voidaan kirjata lomakepohjaisesti tai ne voidaan kerätä potilastietojärjestelmästä.

Jotta terveyskeskustyöyksiköiden potilaiden voimavarakäytön eroja (ns. case mix -tarkastelu) voitaisiin vertailla, täytyy terveyskeskuksen toimintojen (ATL-luokkien) olla arvotettuna (painokertoimet/kustannuspainot määriteltynä). Kustannuspainot ovat DRG Medical Systems Oy:n vastuulla (myös jatkossa, vaikka itse ryhmittelyn omistaisikin joku muu). Kustannuspainot muodostuvat seuraavasti:

- 1) Ensimmäisessä tasossa jokaiselle ATL-luokituksen toimenpiteelle on määritetty painokerroin sen mukaan, minkä verran työaikaa, avustavan henkilökunnan työaikaa ja mahd. materiaalikuluja niihin keskimäärin syntyy.
- 2) Toisella tasolla jokaisen pAPR:n muodostamisen yhteydessä kullekin syntyneelle pAPR:lle lasketaan toimintojen painokertoimien summana kustannuspaino, joka siis kuvaa ko. toimintayksikön (esim. terveyskeskuksen tai terveysaseman) omaa juuri näihin potilaisiin kohdistuvaa toimenpiteiden kustannuspainoa. Tämä laskenta siis tehdään jokaisella ryhmittelykerralla.
- 3) Näiden terveyskeskusten omien kustannuspainojen keskiarvoista muodostetaan lisäksi nk. "valtakunnalliset kustannuspainot", jotka toimivat jatkossa muuttumattomina vertailustandardeina. (Elonheimo 2006)

Palvelun käyttöönotto asiakassovelluksessa edellyttää asiakaskohtaisista tietosisällön laajennuksista (ks. luku 6.1) ja käytettävistä koodistojen versioista sopimista asiakasrajapinnan toteuttajan (palvelua käyttävän ohjelmiston toimittajan) ja palvelua tarjoavan ohjelmiston toimittajan kesken:

3.2 Tekniikkaan liittyvät vaatimukset

Tässä dokumentissa rajapintaratkaisu on esitetty WSDL/SOAP-tekniikalla. Tämä tarkoittaa rajapintojen kuvaamista Web Services Description Language -kielellä ja rajapintojen käyttämistä Simple Object Access Protocol -protokollalla. Molemmat kielet ovat XML-pohjaisia, ja ne muodostavat

yhdessä infrastruktuuritoimittajien suunnalta tuetuimman sovelluspalvelutekniikan hajautettujen Internet-pohjaisten ohjelmistorajapintojen toteuttamiseen.

WSDL ja SOAP ovat alustariippumattomia ja avoimia tekniikoita, joten tässä dokumentissa kuvatut rajapinnat voidaan toteuttaa moneen ympäristöön ja eri tekniikoilla (kuten .NET ja J2EE) tehtyihin ohjelmistoihin. APR-ryhmittely on näin liitettävissä uusiin ja vanhoihin sovelluksiin verkkopalveluna.

Rajapinnan toteuttamiseen osallistuvien organisaatio-osapuolien tulee toteuttaa WSDL/SOAP-sovellusinfrastruktuuri. Yhteentoimivuuden parantamiseksi toteutuksessa tulisi noudattaa WS-I Basic Profile 1.0 -säännöstöä (SerAPI 2004).

4 Rajapintaratkaisun kuvaus

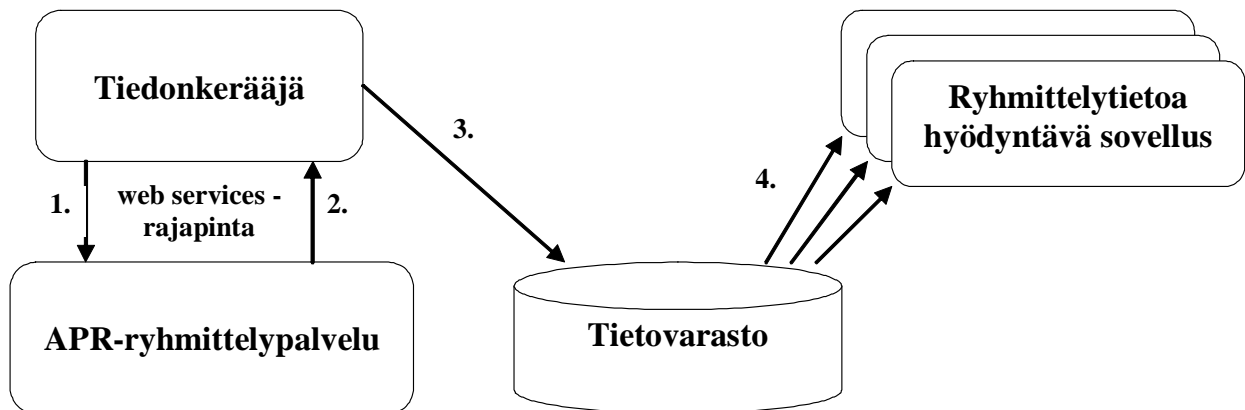
Tässä luvussa kuvataan APR-rajapinnan perusratkaisu ja periaatteet kahdesta eri arkkitehtuurisesta näkökulmasta: yhteistoiminnallisuuden arkkitehtuuri ja tekninen arkkitehtuuri.

4.1 Sovellusosapuolet ja yhteistoiminnallisuuden arkkitehtuuri

Kuvassa 4 on kuvattu APR-ryhmittelyn sekä ryhmittelytulosten käsittelyn ja hyödyntämisen sovellusosapuolet sekä niiden välinen yhteistoiminnallisuus. Tiedonkerääjä kerää APR-ryhmittelyyn tarvittavat tiedot ja 1) lähettää palvelupyynnön APR-ryhmittelypalvelulle. 2) APR-ryhmittelypalvelu lähettää vastauksena ryhmittelyn tulokset, jotka 3) tiedonkerääjä tallentaa tietovarastoon. 4) Ryhmittelytietoa hyödyntävät sovellukset voivat hakea tietovarastosta APR-ryhmittelyn tuloksia, joita ne käyttävät:

- potilasmateriaalin ja sen vaatavuuden kuvaamiseen vakioinnin avulla (case mix -tarkastelu)
- toimintakäytäntöjen kuvaamiseen
- tuottavuus- ja tehokkuuslaskelmiin
- väestön palvelujen tarpeen arviointiin sekä
- alueellisten hoitoketjujen kuvaamiseen (perusterveydenhuolto + erikoissairaanhoido) (DMS 2006d).

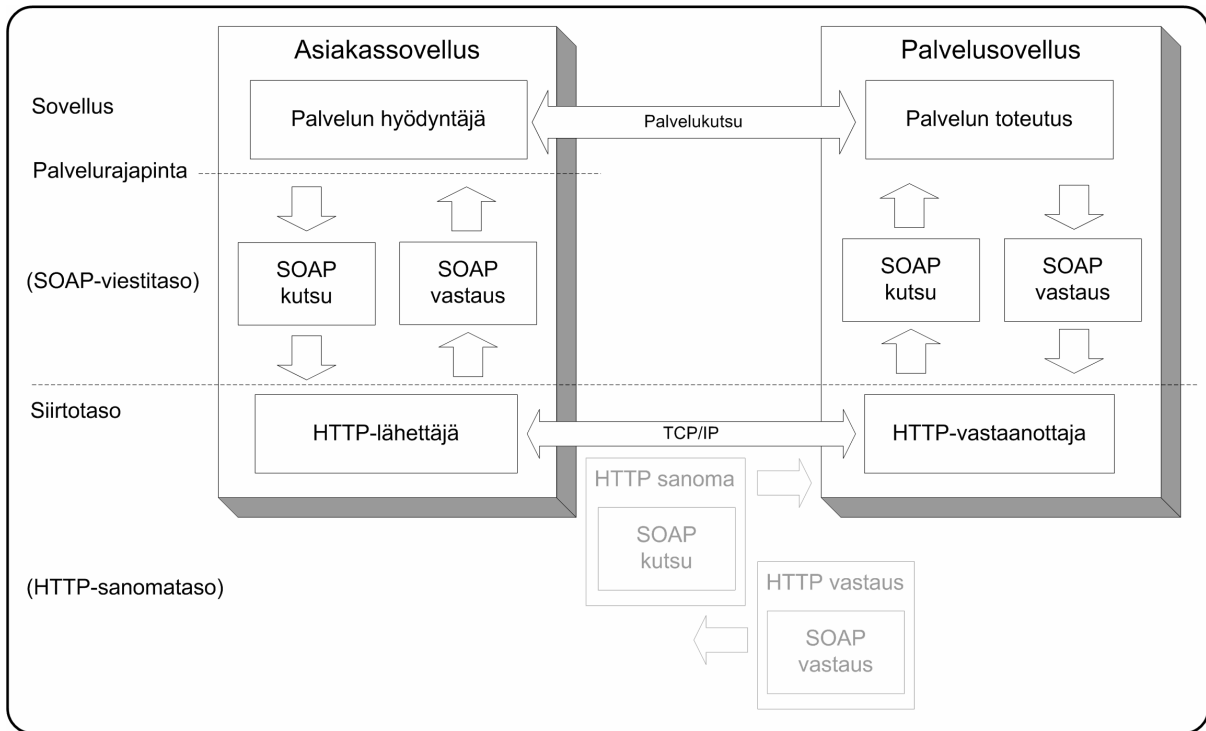
Tiedonkerääjänä voi toimia potilastietojärjestelmä tai esim. erillinen lomakesovellus APR-tiedonkeruuseen. Tietovarasto voi sisältyä tähän sovellukseen, ja myös ryhmittelytietoa hyödyntävä sovellus voi olla sama kuin tiedonkerääjä. Kuvassa 4 nämä roolit on erotettu toisistaan, koska ne voivat myös olla eri ohjelmistojen toteuttamia. APR-ryhmittelypalvelu on itse asiassa rajapinta, jonka takana on APR-ryhmittelijä. Tässä dokumentissa kuvataan tiedonkerääjän ja APR-ryhmittelypalvelun vastuut sekä niiden välinen rajapinta (1. ja 2.).



Kuva 4. APR-ryhmittelyn yhteistoiminnallisuus (nuolet kuvaavat tiedon kulkua)

4.2 Tekninen arkkitehtuuri

Ryhmittelijää käyttävän sovelluksen ja ryhmittelypalvelun välinen tekninen arkkitehtuuri on tyypillinen web services -arkkitehtuuri (ks. kuva 5), jossa käytetään WSDL-rajapintakuvauksia ja HTTP-protokollan yli tapahtuvaa SOAP-sanomavälitystä. Kuvan asiakassovelluksena voi siis toimia perusjärjestelmä tai tiedonkeruusovellus, jonka on huolehdittava myös palvelukutsun tuloksen käsitte-lystä (esim. ryhmittelyn tuloksen tallentamisesta tai välittämisestä edelleen suunnittelun tai johdon tietojärjestelmään). Tässä arkkitehtuurissa ei kuvata ryhmittelytiedon jatkohyödyntämistä.



Kuva 5. Tekninen web services -arkkitehtuuri (SerAPI 2005).

5 Toiminnallisuus

5.1 Roolit ja vastuut

Käyttäjä

Käyttäjällä tarkoitetaan henkilöä, joka kirjaa asiakassovellukseen APR:n (perusterveydenhuollon Avohoidon potilasryhmittelyksen) tarvitsemia tietoja, joita ovat kontaktin syy, toiminnot, kontaktityppi, kontaktin luonne, potilaan ikä ja sukupuoli. Käyttäjä voi olla esimerkiksi lääkäri, sairaanhoitaja, terveydenhoitaja tai fysioterapeutti.

Asiakassovellus

Asiakassovellus kerää APR-ryhmittelyyn tarvittavat tiedot, lähettää palvelupyynnön APR-palveluun varsinaista ryhmittelyä varten ja tallentaa tai käsittelee vastauksen. Asiakassovellus voi kerätä ryhmittelyyn tarvittavat tiedot erillisellä lomakkeella tai tiedot voidaan kerätä käyttäjälle näkymättömällä tavalla kontaktin yhteydessä kerättävistä tiedoista.

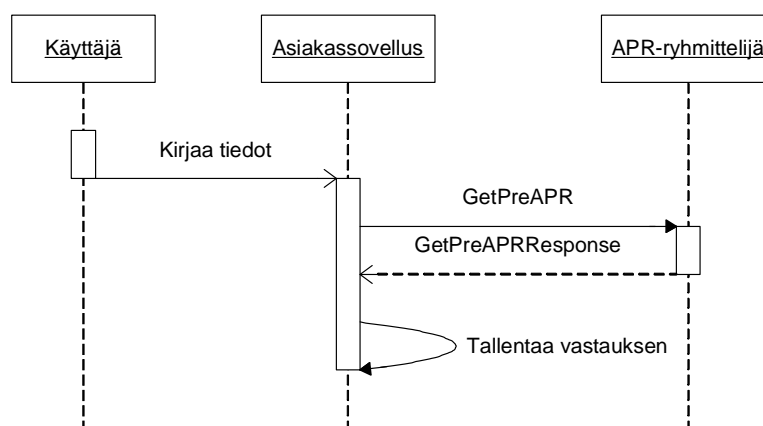
APR-rajapinta (APRService)

Rajapinta, joka välittää saamansa tiedot APR-ryhmittelijälle (sovelluspalvelulle) ja palauttaa asiakassovellukselle ryhmittelijältä saamansa tiedot. APRService tarjoaa Get_pAPR-rajapintaoperaation, joka välittää APR-palvelulle ryhmittelyssä tarvittavat tiedot ja palauttaa pAPR-ryhmän sekä lisätietoja ryhmittelystä. Mikäli ryhmittely ei onnistu, ryhmittelijä palauttaa siitä tiedon.

5.2 Palvelun käyttötavat

Kontaktikohtainen pAPR-ryhmittely.

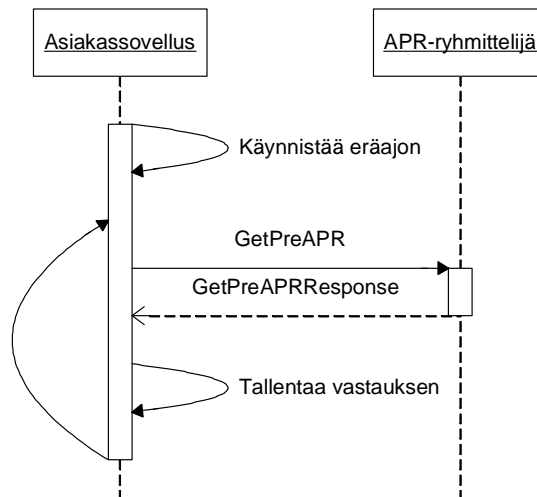
pAPR:n (perusAPR) tallennus tapahtuu reaaliaikaisena, kontaktikohtaisesti. Tällä hetkellä saman kontaktin aikana eri syiden vuoksi tehtyjä toimintoja ei voida erottaa, mutta rajapinnassa on huomioitu mahdollisuus tulevaisuudessa yhdistää saman kontaktin aikana tapahtuneet toiminnot ja syyt toisiaan vastaavalla tavalla.



Kuva 6. Kontaktikohtainen pAPR-ryhmittely

Eräajona tapahtuva pAPR-ryhmittely

APR-rajapinnassa ei ole erillistä operaatiota eräajona tapahtuvaan pAPR-ryhmittelyyn, mutta asiakassovellus (esim. potilastietojärjestelmä) voi halutessaan toistaa useita peräkkäisiä pAPR-ryhmittelyjä ja näin niputtamalla tehdä ryhmittelyn toiminnan kannalta järkevissä kokonaisuuksissa (tunneittain, päivittäin, viikoittain tms.).



Kuva 7. Eräajona tapahtuva pAPR-ryhmittely

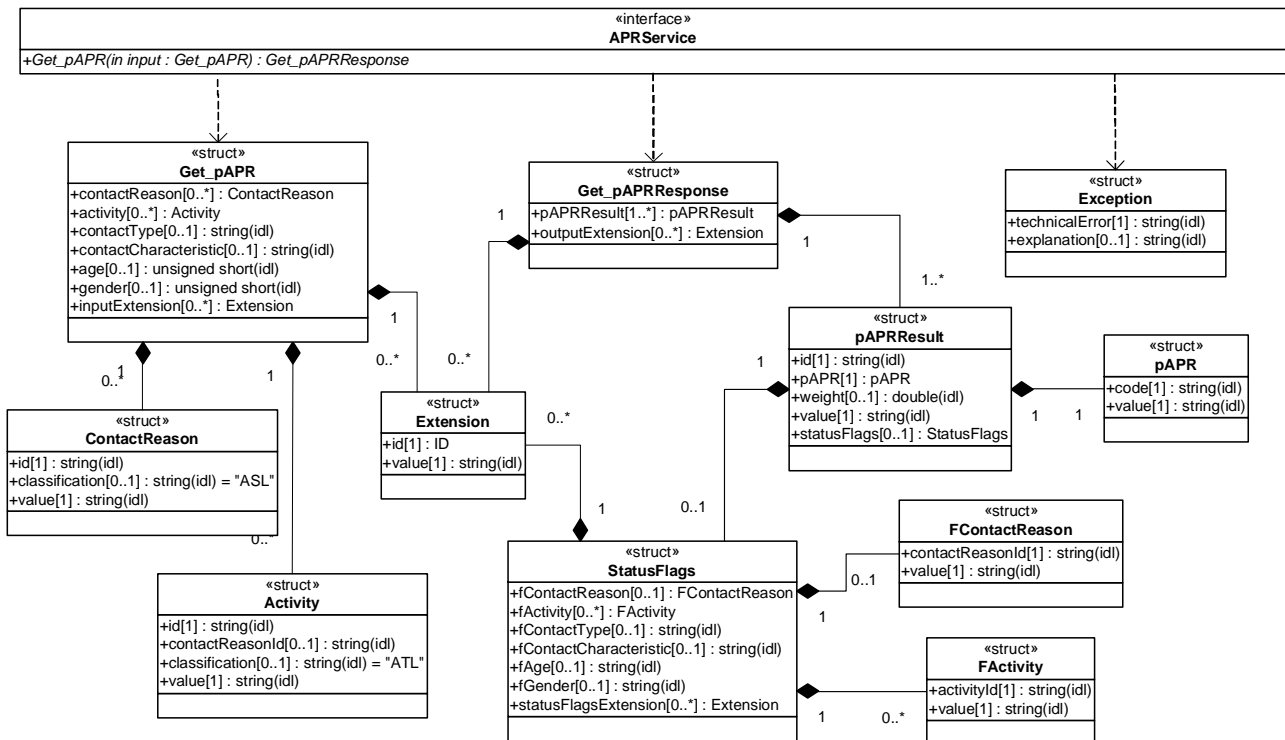
Hoitoepisodikohtainen APR-ryhmittely

Hoitoepisodikohtainen APR-ryhmä (episodiAPR, APR-tuote) syntyy liittämällä yhteen saman tai yhteen liittyvien sairauksien pAPR -ryhmät potilastunnuksen ja sairausluokan mukaan. APR-rajapinnassa ei ole vielä operaatiota episodikohtaiseen ryhmittelyyn, mutta se voidaan lisätä, kun hoitoepisodien käsittelystä sovitaan potilastietojärjestelmissä.

6 Rajapinta ja sen tietosisältö

APR-ryhmittelijälle lähetettävät ja sen palauttavat parametrit - tietoelementit - on määritelty ja nimetty olemassa olevan APR-määrittelyn ja sen käyttöön pohjautuvan toteutuksen pohjalta. Minimitalon toteutus kattaa kaikki onnistuneeseen ryhmittelyyn vähintään tarvittavat tietojoukot. Minimitalon lisäksi määrittelyssä on tunnistettu vapaaehtoisia tietojoukkoja sekä varauduttu mahdollisiin laajennuksiin tietosisällön osalta.

Alla olevassa kaaviossa (kuva 8) on kuvattu APRService-rajapinta (<<interface>>) ja esitelty sen Get_pAPR-operaatio syöteparametreineen ja paluuarvoineen. Parametrit ja paluuarvot on määritelty rakenteisina tietotyyppeinä (<<struct>>). Parametrien ja paluuarvojen yksiarvoiset ominaisuudet kuvataan luokka-attribuutteina, joihin on liitetty tietotyyppi (esim. string) ja attribuutin esiintymisen (1, 0..1, 0..* tai 1..*). Koosteparametrit ja paluuarvot on puolestaan esitetty myös rakenteisina tietotyyppeinä. Kuuluminen toiseen rakenteiseen tietotyyppiin on kuvattu UML-notaation mukaisella koostamissuhteella, eli salmiakkipaisella nuolella.



Kuva 8. APRService-rajapinta ja sen tietotyypit

Seuraavissa alaluvuissa on kuvattu tarkemmin kukin syöteparametri ja paluuarvo sekä koodistojen osalta määritelty käytettävät arvot. Kukin osio sisältää kyseisen tiedon tarpeen, merkityksen ja muodon määrittelyn ja lisäksi teknisen WSDL-kuvauksen.

6.1 Minimitason tietosisältö ja laajennukset

6.1.1 Minimitaso

Minimitasolla kutsun täytyy sisältää:

- (ainakin yksi) kontaktin syy
- (ainakin yksi) toiminto
- kontaktityyppi
- kontaktin luonne
- potilaan ikä
- potilaan sukupuoli

Minimitasolla vastauksen täytyy sisältää:

- (ainakin yksi) pAPR-ryhmä

6.1.2 Laajennukset

Minimitason laajennus voi kattaa kaikki määrittelyssä tunnistetut tietojoukot (ks. kuva 8, luvut 6.2 ja 6.3). Lisäksi on mahdollistettu määrittelyssä tunnistamattomat laajennukset kutsun (*inputExtension*), vastauksen (*outputExtension*) ja vastauksen tilalippujen (*statusFlagsExtension*) tietosisällölle, joiden avulla voidaan lähettää myös tarvittavia tässä määrittelyssä tunnistamattomia tietoja.

6.2 Kutsu (Get_pAPR)

Get_pAPR-kutsu liittyy yhteen kontaktiin ja sen tietosisältöön kuuluvat luvussa 6.1.1 kuvatut minimitason tiedot (poikkeuksena minitasoon, että kaikki syötetiedot on käytännön syistä määritelty vapaaehtoisiksi). Lisäksi on varauduttu mahdollisiin laajennuksiin (*inputExtension*), joita voi olla nolla tai useampia. Alla kuvataan tietosisältö WSDL-muodossa sekä eri parametrien merkitys tarkemmin.

WSDL/SOAP:

Kutsu on määritelty GetPreAPR-elementiksi, jonka sisältö on:

- nolla tai useampi *contactReason*-elementti - kontaktin syy
- nolla tai useampi *activity*-elementti - kontaktin syyhyn liittyvä toiminto
- (nolla tai) yksi *contactType*-elementti - kontaktin tyyppi
- (nolla tai) yksi *contactCharacteristic*-elementti - kontaktin luonne
- (nolla tai) yksi *age*-elementti - potilaan ikä
- (nolla tai) yksi *gender*-elementti - potilaan sukupuoli
- nolla tai useampi *inputExtension*-elementti - laajennukset

Ote WSDL-dokumentista:

```
<element name="Get_pAPR">
  <complexType>
    <sequence>
```

```
<element name="contactReason" type="serapi:ContactReason" minOccurs="0"
  maxOccurs="unbounded" />
<element name="activity" type="serapi:Activity" minOccurs="0"
  maxOccurs="unbounded" />
<element name="contactType" type="string" minOccurs="0" />
<element name="contactCharacteristic" type="string" minOccurs="0" />
<element name="age" type="serapi:unsignedShort" minOccurs="0" />
<element name="gender" type="serapi:unsignedShort" minOccurs="0" />
<element name="inputExtension" type="serapi:Extension" minOccurs="0"
  maxOccurs="unbounded" />
</sequence>
</complexType>
</element>
```

6.2.1 Kontaktin syy (contactReason)

Kontaktin syiden kirjaamiseen käytetään avohoidon sairausluokitusta (ASL). ASL:n sijasta voidaan käyttää myös ICD-10- tai ICPC-diagnoosiluokituksia, jolloin APR-luokittelijan täytyy osata tehdä muunnokset ASL-luokiksi.

WSDL/SOAP:

Kontaktin syyt on määritelty *contactReason*-elementeiksi, joita voi olla nolla tai useampia. Elementti sisältää string-tyyppisen kontaktin syyn tunnusteen, esim. "SL10". Elementillä on kaksi attribuuttia: *id* ja *classification*. Attribuutti *id* on pakollinen ja sen arvo on ID-tyyppinen kontaktin syyn yksilöllinen tunniste (suositeltavat arvot: "cr1", "cr2", "cr3", ...). Attribuutti *classification* on vapaaehtoinen ja sen arvo on string-tyyppinen käytetyn luokituksen tunniste ("ASL", "ICD-10" tai "ICPC"). Attribuutin *classification* oletusarvo on "ASL".

Ote WSDL-dokumentista:

```
<element name="contactReason" type="serapi:ContactReason" minOccurs="0"
  maxOccurs="unbounded" />
<complexType name="ContactReason">
  <simpleContent>
    <extension base="string">
      <attribute name="id" type="ID" use="required" />
      <attribute name="classification" type="string" use="optional"
        default="ASL" />
    </extension>
  </simpleContent>
</complexType>
```

6.2.2 Toiminto (activity)

Toimintojen kirjaamiseen käytetään avohoidon toimintoluokitusta (ATL). Toiminnot liittyvät aina johonkin kontaktin syistä (ks. luku 6.2.1). ATL:n sijasta voidaan käyttää myös NCSP-toimenpideluokitusta (vaatii laajennusta nykyiseen NCSP-luokitukseen), jolloin APR-luokittelijan täytyy osata tehdä muunnokset ATL-luokiksi.

WSDL/SOAP:

Toiminnot on määritelty *activity*-elementeiksi, joita voi olla nolla tai useampia. Elementti sisältää string-tyyppisen toimintokoodin, esim. "TL120". Elementillä on kolme attribuuttia: *id*, *contactReasonId* ja *classification*. Attribuutti *id* on pakollinen ja sen arvo on ID-tyyppinen toiminnon yksilöllinen tunniste (suositeltavat arvot: "ac1", "ac2", "ac3", ...). Attribuutti *contactReasonId* on vapaaehtoinen ja sen arvo on sen kontaktin syyn tunniste, johon toiminto liittyy. Attribuuttia *contactReasonId* ei tarvitse käyttää, jos kutsu sisältää vain yhden kontaktin syyn tai toimintoa ei pystytä liittämään tiettyyn kontaktin syyhyn. Attribuutti *classification* on vapaaehtoinen ja sen arvo on käytetyn luokituksen tunniste ("ATL" tai "NCSP"). Attribuutin *classification* oletusarvo on "ATL".

Ote WSDL-dokumentista:

```
<element name="activity" type="serapi:Activity" minOccurs="0"
  maxOccurs="unbounded" />

<complexType name="Activity">
  <simpleContent>
    <extension base="string">
      <attribute name="id" type="ID" use="required" />
      <attribute name="contactReasonId" type="string" use="optional" />
      <attribute name="classification" type="string" use="optional"
        default="ATL" />
    </extension>
  </simpleContent>
</complexType>
```

6.2.3 Kontaktityyppi (contactType)

Kontaktityypin kirjaamiseen käytetään kontaktityyppi-luokitusta.

WSDL/SOAP:

Kontaktityyppi on määritelty *contactType*-elementiksi, jonka sisältö on string-tyyppinen kontaktityypin koodi, esim. "KL10".

Ote WSDL-dokumentista:

```
<element name="contactType" type="string" minOccurs="0" />
```

6.2.4 Kontaktin luonne (contactCharacteristic)

Kontaktin luonteen kirjaamiseen käytetään kontaktin luonne -luokitusta.

WSDL/SOAP:

Kontaktin luonne on määritelty *contactCharacteristic*-elementiksi, joka sisältää string-tyyppisen kontaktin luonteen koodin, esim. "LU10".

Ote WSDL-dokumentista:

```
<element name="contactCharacteristic" type="string" minOccurs="0" />
```

6.2.5 Potilaan ikä (age)

Potilaan ikä (vuosina, kokonaisluku) voidaan päätellä potilastietojärjestelmässä henkilötunnuksesta ja käyntipäivästä.

WSDL/SOAP:

Potilaan ikä on määritelty *age*-elementiksi, jonka sisältö on unsignedShort-tyyppinen potilaan ikä vuosina.

Ote WSDL-dokumentista:

```
<element name="age" type="serapi:unsignedShort" minOccurs="0"/>
```

6.2.6 Potilaan sukupuoli (gender)

Potilaan sukupuoli voidaan päätellä potilastietojärjestelmässä henkilötunnuksesta. Arvona (etumerkitön kokonaisluku) käytetään HL7 v2.3 -sukupuoli-koodeja (koodiston OID 1.2.246.777.5.1.1):

Arvo	Selitys
1	male, mies
2	female, nainen
3	unknown, ei tiedossa/määriteltävissä

WSDL/SOAP:

Potilaan sukupuoli on määritelty *gender*-elementiksi, jonka sisältö on unsignedShort-tyyppinen potilaan sukupuolen HL7 v2.3 -tyyppikoodi.

Ote WSDL-dokumentista:

```
<element name="gender" type="serapi:unsignedShort" minOccurs="0"/>
```

6.2.7 Kutsun laajennus (inputExtension)

Kutsun laajennus sisältää seuraavat tiedot:

- laajennuksen yksilöllinen tunniste (*id*) merkkijonona (tai esimerkiksi XML-skeemassa ID-tyyppinä)
- laajennuksen sisältö merkkijonona

WSDL/SOAP:

Kutsun laajennus on määritelty *inputExtension*-elementiksi. Elementillä on yksi attribuutti: *id*, joka on pakollinen ja jonka arvo on ID-tyyppinen laajennuksen yksilöllinen tunniste. Elementin sisältö on string-tyyppinen laajennuksen arvo.

Ote WSDL-dokumentista:

```
<element name="inputExtension" type="serapi:Extension" minOccurs="0"
  maxOccurs="unbounded" />
```

```
<complexType name="Extension">
  <simpleContent>
    <extension base="string">
      <attribute name="id" type="ID" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

6.3 Vastaus (Get_pAPRResponse)

Operaation Get_pAPR kutsuminen palauttaa onnistuneen pAPR-ryhmittelyn tulokset tai virheilmoituksen. Onnistuneen luokittelukutsun tietosisältöön kuuluu:

- yksi tai useampi pAPR-ryhmittelyn tulos (yksi jokaista kutsun sisältämää kontaktin syytä kohti)

Lisäksi on varauduttu mahdollisiin laajennuksiin (*outputExtension*). Laajennuksia voi olla nolla tai useampi.

WSDL/SOAP:

Vastaus kutsuun on määritelty Get_pAPRResponse-elementiksi, jonka sisältö on:

- yksi tai useampi *pAPRResult*-elementti
- nolla tai useampi *outputExtension*-elementti

Ote WSDL-dokumentista:

```
<element name="Get_pAPRResponse">
  <complexType>
    <sequence>
      <element name="pAPRResult" type="serapi:pAPRResult"
        maxOccurs="unbounded" />
      <element name="outputExtension" type="serapi:Extension" minOccurs="0"
        maxOccurs="unbounded" />
    </sequence>
  </complexType>
</element>
```

6.3.1 pAPR-ryhmittelyn tulos (pAPRResult)

Vastaus sisältää kutakin kutsun sisältämää kontaktin syytä kohti yhden pAPR-ryhmittelyn tuloksen, joka sisältää seuraavat tiedot:

- pAPR-ryhmittelyn tuloksen yksilöllinen tunniste (*id*)
 - arvo merkkijono (tai esimerkiksi XML-skeemassa ID-tyyppi)
 - suositeltavat arvot: "re1", "re2", "re3", ...
- pAPR-ryhmä (*pAPR*, ks. luku 6.3.2)
- mahdollisesti pAPR-ryhmän painokerroin (kustannuspaino) (*weight*, ks. luku 6.3.3)
 - arvo desimaaliluku
- mahdollisesti pAPR-ryhmittelyn tulokseen liittyvät tilaliput (*statusFlags*, ks. luku 6.3.4)
 - ilmaisevat syötetietojen validiutta ja käyttöä ryhmittelyssä

WSDL/SOAP:

pAPR-ryhmittelyn tulokset on määritelty *pAPRResult*-elementeiksi. Elementillä on yksi attribuutti: *id*, joka on pakollinen ja jonka arvo on ID-tyyppinen pAPR-ryhmittelyn tuloksen yksilöllinen tunniste (suositeltavat arvot: "re1", "re2", "re3", ...). Elementin sisältö on yksi pAPR-elementti (ks. luku 6.3.2), nolla tai yksi weight-elementti (ks. luku 6.3.3) ja nolla tai yksi statusFlags-elementti (ks. luku 6.3.4).

Ote WSDL-dokumentista:

```
<element name="pAPRResult" type="serapi:pAPRResult" maxOccurs="unbounded"/>

<complexType name="pAPRResult">
  <sequence>
    <element name="pAPR" type="serapi:pAPR"/>
    <element name="weight" type="decimal" minOccurs="0"/>
    <element name="statusFlags" type="serapi:StatusFlags" minOccurs="0"/>
  </sequence>
  <attribute name="id" type="ID" use="required"/>
</complexType>
```

6.3.2 pAPR-ryhmä (pAPR)

pAPR-ryhmä sisältää seuraavat tiedot:

- pAPR-ryhmän koodi (*code*) merkkijonona
- pAPR-ryhmän selite merkkijonona

WSDL/SOAP:

pAPR-ryhmä on määritelty *pAPR*-elementiksi. Elementillä on yksi attribuutti: *code*, joka on pakollinen ja jonka arvo on string-tyyppinen pAPR-ryhmän koodi (tunniste), esim. "SL10_B". Elementin sisältö on string-tyyppinen pAPR-ryhmän selite, esim. "Verenpainetaudin tavanomainen ohjaus ja hoito".

Ote WSDL-dokumentista:

```
<element name="pAPR" type="serapi:pAPR"/>

<complexType name="pAPR">
  <simpleContent>
    <extension base="string">
      <attribute name="code" type="string" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

6.3.3 pAPR-ryhmän painokerroin (weight)

pAPR-ryhmän painokerroin (kustannuspaino), arvo desimaaliluku.

WSDL/SOAP:

pAPR-ryhmän painokerroin on määritelty *weight*-elementiksi, jonka sisältö on decimal-tyyppinen pAPR-ryhmän painokerroin, esim. "1.5".

Ote WSDL-dokumentista:

```
<element name="weight" type="decimal" minOccurs="0"/>
```

6.3.4 Tilaliput (statusFlags)

Kuhunkin pAPR-ryhmään voi liittyä tilaliput, joiden avulla kuvataan, onko eri syötetietoja käytetty ryhmittelyssä ja ovatko niiden arvot olleet valideja ryhmittelyn kannalta. Tilalippujen tietosisältöön kuuluu:

- nolla tai useampi kontaktin syy -tilalippu (*fContactReason*)
 - liitetään kutsussa olleeseen kontaktin syyhyn yksilöllisellä tunnisteella (*contactReasonId*)
- nolla tai useampi toiminto-tilalippu (*fActivity*)
 - liitetään kutsussa olleeseen toimintoon yksilöllisellä tunnisteella (*activityId*)
- mahdollinen kontaktityyppi-tilalippu (*fContactType*)
- mahdollinen kontaktin luonne -tilalippu (*fContactCharacteristic*)
- mahdollinen ikä-tilalippu (*fAge*)
- mahdollinen sukupuoli-tilalippu (*fGender*)

Lisäksi on varauduttu mahdollisiin tilalippulaajennuksiin (*statusFlagsExtension*). Laajennuksia voi olla nolla tai useampi.

Kunkin tilalipun arvo on merkkijono, joka sisältää kaksi numeerista merkkiä:

- 1. merkki:

Arvo	Selitys
0	Syötetiedon arvoa ei ole käytetty ryhmittelyyn
1	Syötetiedon arvoa on käytetty ryhmittelyyn

- 2. merkki:

Arvo	Selitys
0	Syötetiedon arvo ei ole validi
1	Syötetiedon arvo on validi

WSDL/SOAP:

Yhteen pAPR-ryhmään liittyvät tilaliput on koottu *statusFlags*-elementtiin, joka voi esiintyä nolla tai useamman kertaa. Elementin sisältö on:

- nolla tai useampi *fContactReason*-elementti
 - elementillä on yksi pakollinen attribuutti: *contactReasonId*, jonka arvo on sen kutsussa olleen kontaktin syyn yksilöllinen tunniste, johon tämä kontaktin syy -tilalippu liittyy
- nolla tai useampi *fActivity*-elementti

- elementillä on yksi pakollinen attribuutti: *activityId*, jonka arvo on sen kutsussa olleen toiminnon yksilöllinen tunniste, johon tämä toiminto-tilalippu liittyy
- mahdollinen *fContactType*-elementti
- mahdollinen *fContactCharacteristic*-elementti
- mahdollinen *fAge*-elementti
- mahdollinen *fGender*-elementti
- nolla tai useampi *statusFlagsExtension*-elementti

Kaikkien tilalippuelementtien sisältö on string-tyyppinen tilalipun arvo.

Ote WSDL-dokumentista:

```
<element name="statusFlags" type="serapi:StatusFlags" minOccurs="0"/>

<complexType name="StatusFlags">
  <sequence>
    <element name="fContactReason" type="serapi:FContactReason" minOccurs="0"/>
    <element name="fActivity" type="serapi:FActivity" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="fContactType" type="string" minOccurs="0"/>
    <element name="fContactCharacteristic" type="string" minOccurs="0"/>
    <element name="fAge" type="string" minOccurs="0"/>
    <element name="fGender" type="string" minOccurs="0"/>
    <element name="statusFlagsExtension" type="serapi:Extension" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="FContactReason">
  <simpleContent>
    <extension base="string">
      <attribute name="contactReasonId" type="string" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<complexType name="FActivity">
  <simpleContent>
    <extension base="string">
      <attribute name="activityId" type="string" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

6.3.4.1 Tilalippulaajennus (statusFlagsExtension)

Tilalippulaajennus sisältää seuraavat tiedot:

- laajennuksen yksilöllinen tunniste (*id*) merkkijonona (tai esimerkiksi XML-skeemassa ID-tyyppinä)
- laajennuksen sisältö merkkijonona

WSDL/SOAP:

Tilalippulaajennus on määritelty *statusFlagsExtension*-elementiksi. Elementillä on yksi attribuutti: *id*, joka on pakollinen ja jonka arvo on ID-tyyppinen laajennuksen yksilöllinen tunniste. Elementin *statusFlagsExtension* sisältö on string-tyyppinen laajennuksen arvo.

Ote WSDL-dokumentista:

```
<element name="statusFlagsExtension" type="serapi:Extension" minOccurs="0"
  maxOccurs="unbounded" />

<complexType name="Extension">
  <simpleContent>
    <extension base="string">
      <attribute name="id" type="ID" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

6.3.5 Vastauksen laajennus (outputExtension)

Vastauksen laajennus, joka sisältää tiedot:

- laajennuksen yksilöllinen tunniste (*id*) merkkijonona (tai esimerkiksi XML-skeemassa ID-tyyppinä)
- laajennuksen sisältö merkkijonona

WSDL/SOAP:

Vastauksen laajennus on määritelty *outputExtension*-elementiksi. Elementillä on yksi attribuutti: *id*, joka on pakollinen ja jonka arvo on ID-tyyppinen laajennuksen yksilöllinen tunniste. Elementin sisältö on string-tyyppinen laajennuksen arvo.

Ote WSDL-dokumentista:

```
<element name="outputExtension" type="serapi:Extension" minOccurs="0"
  maxOccurs="unbounded" />

<complexType name="Extension">
  <simpleContent>
    <extension base="string">
      <attribute name="id" type="ID" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

6.4 Virhetilanteet

APR-luokittelija voi asiakasohjelman tekemän palvelupyynnön suorittamisen aikana ajautua virhetilanteeseen. Tällöin palautetaan virheilmoitus (*Exception*), joka sisältää seuraavat tiedot:

- virheen tyyppi (*type*) merkkijonona
- mahdollisesti virheen tarkempi kuvaus (*explanation*) merkkijonona

Virhetyyppejä ovat:

- *GeneralFailure* – erittelemätön virhe
- *MissingInputParameter* - pakollinen tieto puuttuu
- *ClassificationNotSupported* - annettua luokitusta ei tueta
- *UnknownCode* - annettua koodia ei löydy luokituksista
- *UnknownInputExtension* – ryhmittelijä ei tunnistanut käytettyä laajennusta

WSDL/SOAP:

Virhetilanteessa luodaan SoapException. Itse virheilmoitus on määritelty *Exception*-elementiksi SoapException:n Detail-osaan. Elementti *Exception* sisältää yhden *type-elementin* ja nolla tai yksi *explanation-elementtiä*. Elementti *type* sisältää virhetyypin merkkijonona. Elementti *explanation* sisältää virheen tarkemman kuvauksen merkkijonona.

Ote WSDL-dokumentista:

```
<element name="Exception">
  <complexType>
    <sequence>
      <element name="type" type="string"/>
      <element name="explanation" type="string" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
```

6.5 Esimerkkejä SOAP-viesteistä

Tässä luvussa esitellään esimerkkejä SOAP-sanomista, joilla asiakasprosessi voi kutsua palveluteutuksen APR-rajapintaa, sekä SOAP-sanomia, jotka palvelutoteutus lähettää vastauksena kutsuun.

Esimerkki 1. SOAP-palvelupyynnö (Get_pAPR, minimitaso)

```
POST /APRService/APRService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "urn:serapi:APR#Get_pAPR"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Get_pAPR xmlns="urn:serapi:APR">
      <contactReason id="cr1">SL10</contactReason>
      <activity id="ac1">TL120</activity>
      <contactType>KL10</contactType>
      <contactCharacteristic>LU10</contactCharacteristic>
      <age>30</age>
      <gender>1</gender>
    </Get_pAPR>
  </soap:Body>
</soap:Envelope>
```

Esimerkki 2. SOAP-vastaus (Get_pAPRResponse, minimitaso)

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Get_pAPRResponse xmlns="urn:serapi:APR">
      <pAPRResult id="rel">
        <pAPR code="SL10_B">xxx</pAPR>
      </pAPRResult>
    </Get_pAPRResponse>
  </soap:Body>
</soap:Envelope>
```

7 Toteutuskohtaiset asiat

Tässä luvussa käydään läpi rajapintatoteutusten laatuun ja tehokkuuteen liittyviä asioita. Lisäksi rajapinnan toteuttajien ja hyödyntäjien avuksi käydään läpi asioita, joiden avulla rajapintojen toteuttaja tai hyödyntäjä voi halutessaan varmentaa, vastaako toteutus tätä määritystä. Toteutuskohtaisesti tulee myös huomioida vaadittava tietoturvan taso ja sen saavuttamiseksi tarvittavat standardit (esim. HTTPS, WS-I Basic Security Profile).

7.1 Rajapinnan toteutuksessa huomioitavat asiat

Asiakassovelluksen vastuulla on huolehtia siitä, että tiedot välitetään APR-ryhmittelijälle määritellyssä muodossa, esimerkkinä tietojen oikeellisuus ja muoto sekä koodistojen mukaisuus (esim. ASL).

Palvelun tarjoaja dokumentoi tuote- ja versio- tai asennuskohtaisesti mm. seuraavat asiat:

- palvelun osoite
- mitä koodistoja ja niiden versioita palvelu edellyttää käytettäväksi ja tukee kutsuissa, ja miten toimitaan asiakassovellukselta vaadittavien koodistojen suhteen esim. ryhmittelijä- tai tuoteversion muuttuessa
- asetetaanko palvelun käyttäjäsovelluksille lisärajoitteita tai -vaatimuksia esim. ryhmittelyn massiivisen toistuvan käytön tai muiden palvelun tarjoajaa runsaasti kuormittavien toimintojen suhteen
- vaatiiko palvelu asiakassovelluksilta tunnistautumis- tai lisensointimekanismien toteuttamista. Tarvittaessa tunnistus ja yksilöinti voidaan toteuttaa esim. PlugIT-projektin käyttäjärajapinnan määritysten mukaisesti (PlugIT 2004b)
- käytössä olevat laajennukset (ks. luvut 6.2.7, 6.3.4.1 ja 6.3.5)
- aiheuttaako palvelulle tuntematon kutsun laajennus (inputExtension, ks. luku 6.2.7) aina virhetilanteen (UnknownInputExtension, ks. virhetilanteet luku 6.4)

Erillisissä palvelutasosopimuksissa tulisi lisäksi kuvata mm.:

- vastuut palvelun tarjoajan ollessa esim. teknisen vian vuoksi saavuttamattomissa
- tarjottavat tukipalvelut (esim. tekninen neuvonta, virheiden korjaus)

7.2 Turvallisuuskohdat

Asiakassovelluksen ja palvelun välisessä kommunikoinnissa ei siirretä tietoja, joiden perusteella tiedot voitaisiin suoraan yhdistää tiettyyn henkilöön.

7.3 Määrittelyn mukaisuus

Kohteen rajapintojen toteuttaja tai hyödyntäjä voi halutessaan varmentaa toteutuksen ja tämän määrittelyn vastaavuuden. Tässä luvussa kerrotaan, mitä kohtia varmentamisessa on käytävä läpi.

Pyynnöissä ja vastauksissa käytettyjen parametrien järjestyksen tulee olla tämän dokumentin mukainen. Lisäksi rajapintaa tarjoavien ja käyttävien sovellusten on noudatettava tämän dokumentin mukaisia standardeja ja koodistoja.

Palvelun toteutus on tämän määrittelyn mukainen, jos sen toiminta ja palauttavat arvot ovat määrittelyn mukaisia. Tämä todetaan testausraportista, joka on tuotettu joko kehitys- tai käyttöympäristössä. Hyväksymistestauksen suorittaminen on suositeltavaa käyttöympäristössä.

Palveluja käyttävän sovelluksen määrittelyjen mukaisuus voidaan todeta testausraportilla, jossa palvelun onnistuneiden kutsujen lisäksi selviää palvelua käyttävän sovelluksen toiminta palvelun palauttaessa virhetilanteita tai ollessa saavuttamattomissa.

8 Määrityksen jatkokehitys ja toteutukset

Tämä määrittäminen on tehty siten, että sen pohjalta voidaan toteuttaa avoimia rajapintoja ohjelmistoihin ja sitä voidaan käyttää esim. julkisissa tarjouspyynnöissä, kun se on julkisesti saatavilla.

Määrittäminen voidaan julkistaa, hyväksyä standardiksi tai ottaa pohjaksi kehitettäessä uusia versioita SerAPI-projektissa. Uusien versioiden kehittäminen ja standardointi ovat mahdollisia myös julkistetun määrityksen pohjalta projektin ulkopuolella.

Määrityksen käyttökokemuksia ja mahdollisia korjaus-, muutos- ja jatkokehitystarpeita pyydetään välittämään tekijöille. Jatkossa rajapintaan on tarkoitus lisätä tuki (hoito)episoditasoiselle APR-ryhmittelylle ja muuttaa luokitusten tunnisteet OID-koodeiksi (tällä hetkellä ASL- ja ATL-luokituksilla ei ole olemassa OID-koodeja). Rajapintaa on mahdollista laajentaa myös eräajona tapahtuvaa ryhmittelyä varten.

Tätä rajapintamäärittelyä on mahdollista käyttää mallina myös uusille rajapinnoille, jotka voivat liittyä esimerkiksi sairaaloiden hoitotyön arviointijärjestelmään (RAFAELA), vanhustenhuollon arviointi- ja seuranta-järjestelmään (RAI, Resident Assessment Instrument) tai vanhusten pitkäaikashoidon asiakasrakenteen luokittelujärjestelmään (RUG-III, Resource Utilization Groups).

Kuopion yliopiston HIS-tutkimusyksikössä voidaan toteuttaa esimerkki/esimerkkejä APR-ryhmittelyn asiakasovelluksesta .NET- ja/tai Java-tekniikoilla. SerAPI-projektin tutkimusryhmä tukee hankkeen puitteissa toteutuksia eri tietojärjestelmiin, ja myös dokumentissa mainitut yritykset antavat lisätietoja toteutuksista.

Lähteet

DMS 2005. DRG Medical Systems Oy. APR-ryhmittely. Viitattu 13.2.2005.

http://www.apr.fi/APR_ryhmittely_130205.pdf

DMS 2006a. DRG Medical Systems Oy. Viitattu 13.2.2006.

<http://www.apr.fi>

DMS 2006b. APR-Tietojärjestelmä. DRG Medical Systems Oy:n kotisivut. Viitattu 13.2.2006.

<http://www.apr.fi/apr.htm>

DMS 2006c. DRG Medical Systems Oy. Tuotteistus terveyskeskuksessa. 2006.

DMS 2006d. DRG Medical Systems Oy. Avohoidon potilasryhmityksellä toiminnan kuvauksesta tuottavuuteen. APR-esittely. 17.1.2006.

Elonheimo 2006. Sähköposti Outi Elonheimolta. 26.5.2006.

Kuntaliitto 2005a. DRG:n tausta ja DRG-keskuksen toiminta-ajatus, tehtävät ja tavoitteet. Viitattu 7.3.2006.

<http://norddrg.kuntaliitto.fi/2DRGn%20perusteet/DRGn%20tausta%20ja%20DRGkeskus.doc>

Kuntaliitto 2005b. DRG ja NordDRG - Usein kysytyt kysymykset. Viitattu 7.3.2006.

<http://norddrg.kuntaliitto.fi/6usein%20kysytyt%20kysymykset/Kysymys%20ja%20Vastaus.HTM>

NOMESCO. NOMESCO Classification of Surgical Procedures (NCSP), version 1.9. 70:2004. Viitattu 28.3.2006.

http://www.nordclass.uu.se/verksam/Ncsp1_9.pdf

Pessi 2006. Timo Pessi, Datawell Oy. Perusterveydenhuollon avohoidon potilasryhmitys APR - Alustus rajapintojen suunnittelun pohjaksi. Esitys SerAPI-työpajaseminaarissa 9.2.2006. SerAPI-hankkeen sisäistä materiaalia.

PlugIT 2004a. Mykkänen J., Häyrynen K., Savolainen S., Porrasmä J. Standardien arviointi ja valinta terveydenhuollon sovellusintegraatiossa. PlugIT-hankkeen selvityksiä ja raportteja 3. Kuopio, 2004.

<http://www.plugin.fi/julkaisut/03-Mykkanen-ym.pdf>

PlugIT 2004b. Sormunen M., Porrasmä J., Silvennoinen R., Mykkänen J., Savolainen S., Rannanheimo J. Terveystenhuollon avoimet sovellusrajapinnat - käyttäjä- ja käyttöoikeusrajapinnat. PlugIT-hankkeen selvityksiä ja raportteja 9. Kuopio, 2004.

<http://www.plugin.fi/julkaisut/09-Sormunen-ym.pdf>

SerAPI 2004. Yhteenveto WS-I:n kehittämistä määrityksistä. 17.12.2004. SerAPI-hankkeen sisäistä materiaalia.

SerAPI 2005. Web-sovelluspalveluiden teknisiä suosituksia. 18.11.2005.

http://www.uku.fi/tike/his/serapi/mater/WS-suositukset_v1.doc

Stakes 2006. Sosiaali- ja terveydenhuollon käsitteitä tietojärjestelmien suunnittelua varten. Viitattu 28.3.2006.

<http://www.stakes.fi/oske/terminologia/sanastot/kasite.htm>

Liite 1: WSDL-rajapintakuvaus

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace="urn:serapi:APR"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:serapi="urn:serapi:APR"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="APRv1">

  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:serapi="urn:serapi:APR"
      targetNamespace="urn:serapi:APR"
      elementFormDefault="qualified">

      <element name="Get_pAPR">
        <complexType>
          <sequence>
            <element name="contactReason" type="serapi:ContactReason"
              minOccurs="0" maxOccurs="unbounded"/>
            <element name="activity" type="serapi:Activity" minOccurs="0"
              maxOccurs="unbounded"/>
            <element name="contactType" type="string" minOccurs="0"/>
            <element name="contactCharacteristic" type="string"
              minOccurs="0"/>
            <element name="age" type="serapi:unsignedShort" minOccurs="0"/>
            <element name="gender" type="serapi:unsignedShort" minOccurs="0"/>
            <element name="inputExtension" type="serapi:Extension"
              minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>

      <element name="Get_pAPRResponse">
        <complexType>
          <sequence>
            <element name="pAPRResult" type="serapi:pAPRResult"
              maxOccurs="unbounded"/>
            <element name="outputExtension" type="serapi:Extension"
              minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>

      <element name="Exception">
        <complexType>
          <sequence>
            <element name="type" type="string"/>
            <element name="explanation" type="string" minOccurs="0"/>
          </sequence>
        </complexType>
      </element>

      <complexType name="ContactReason">
        <simpleContent>
          <extension base="string">
            <attribute name="id" type="ID" use="required"/>
            <attribute name="classification" type="string" use="optional"
              default="ASL"/>
          </extension>
        </simpleContent>
      </complexType>
    </schema>
  </types>

```

```

        </extension>
    </simpleContent>
</complexType>

<complexType name="Activity">
    <simpleContent>
        <extension base="string">
            <attribute name="id" type="ID" use="required"/>
            <attribute name="contactReasonId" type="string" use="optional"/>
            <attribute name="classification" type="string" use="optional"
                default="ATL"/>
        </extension>
    </simpleContent>
</complexType>

<complexType name="Extension">
    <simpleContent>
        <extension base="string">
            <attribute name="id" type="ID" use="required"/>
        </extension>
    </simpleContent>
</complexType>

<complexType name="pAPRResult">
    <sequence>
        <element name="pAPR" type="serapi:pAPR"/>
        <element name="weight" type="decimal" minOccurs="0"/>
        <element name="statusFlags" type="serapi:StatusFlags"
            minOccurs="0"/>
    </sequence>
    <attribute name="id" type="ID" use="required"/>
</complexType>

<complexType name="pAPR">
    <simpleContent>
        <extension base="string">
            <attribute name="code" type="string" use="required"/>
        </extension>
    </simpleContent>
</complexType>

<complexType name="StatusFlags">
    <sequence>
        <element name="fContactReason" type="serapi:FContactReason"
            minOccurs="0"/>
        <element name="fActivity" type="serapi:FActivity" minOccurs="0"
            maxOccurs="unbounded"/>
        <element name="fContactType" type="string" minOccurs="0"/>
        <element name="fContactCharacteristic" type="string" minOccurs="0"/>
        <element name="fAge" type="string" minOccurs="0"/>
        <element name="fGender" type="string" minOccurs="0"/>
        <element name="statusFlagsExtension" type="serapi:Extension"
            minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>

<complexType name="FContactReason">
    <simpleContent>
        <extension base="string">
            <attribute name="contactReasonId" type="string" use="required"/>
        </extension>
    </simpleContent>
</complexType>

```

```

        </simpleContent>
    </complexType>

    <complexType name="FActivity">
        <simpleContent>
            <extension base="string">
                <attribute name="activityId" type="string" use="required"/>
            </extension>
        </simpleContent>
    </complexType>

</schema>
</types>

<message name="Get_pAPRRequestMessage">
    <part name="parameters" element="serapi:Get_pAPR"/>
</message>
<message name="Get_pAPRResponseMessage">
    <part name="parameters" element="serapi:Get_pAPRResponse"/>
</message>
<message name="ExceptionMessage">
    <part name="fault" element="serapi:Exception"/>
</message>

<portType name="APRServiceOperations">
    <operation name="Get_pAPR">
        <input message="serapi:Get_pAPRRequestMessage"/>
        <output message="serapi:Get_pAPRResponseMessage"/>
        <fault name="Exception" message="serapi:ExceptionMessage"/>
    </operation>
</portType>

<binding name="APRServiceSOAPBinding" type="serapi:APRServiceOperations">
    <wsdlsoap:binding transport="http://schemas.xmlsoap.org/soap/http"
        style="document"/>
    <operation name="Get_pAPR">
        <wsdlsoap:operation soapAction="urn:serapi:APR#Get_pAPR"/>
        <input>
            <wsdlsoap:body use="literal"/>
        </input>
        <output>
            <wsdlsoap:body use="literal"/>
        </output>
        <fault name="Exception">
            <wsdlsoap:fault name="Exception" use="literal"/>
        </fault>
    </operation>
</binding>

<service name="APRService">
    <port name="APRServiceSOAPPort" binding="serapi:APRServiceSOAPBinding">
        <wsdlsoap:address
            location="http://localhost/APRService/services/APRService"/>
    </port>
</service>

</definitions>

```