

OID-generaattorin rajapintamäärittely

Versio 1.0

	SerAPI-projekti	
	Yhteyshenkilö	Esa Paakkanen (Esa.Paakkanen@uku.fi)
	Dokumentin tila	Julkistettu
	Päiväys	5.9.2007

Sisällysluettelo

1	Johdanto.....	4
1.1	Dokumentissa käytetyt tietotekniset ja terveydenhuollon käsitteet	4
2	Rajapinnan kuvaus	6
2.1	Vaatimukset ja rajaukset	6
2.2	Rajapinnan toiminnallisuus	6
2.3	Rajapinnan operaatiot	7
2.3.1	newOid	8
2.3.2	newOidForPerson	8
2.3.3	serverCapability	9
2.4	Virhetilanteet	9
3	OID-tunnuksen ja generaattorin käyttötapoja	11
3.1	Henkilön yksilöinti	11
3.2	Sähköisen asiakirjan ja sen osien yksilöinti	11

Liite 1: OID-generaattorin WSDL-määrittely

Liite 2. Suomalaisen hyödyntämisohjelman mukaiset OID-solmuluokat

Versiohistoria

Versio:	Pvm:	Laatijat:	Selitys:
Versio 1.0	21.11.2005	Esa Paakkanen Saara Savolainen Juha Mykkänen	Julkistettava versio

Esipuhe

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

Tämä dokumentti kuvaa OID-generaattorin tekniikkariippumattomat ja tekniset ratkaisut. Dokumentin tarkoituksena on toimia OID-generaattorin rajapinnan määrittelyä tarjoten sekä rajapintakuvausta että helposti käsiteltävään tietopakettiin generaattorin käytöstä.

1 Johdanto

SerAPI-projektissa on tavoitteena tuottaa tarkasti ja selkeästi määritellyt OID-yksilöinnin käyttöä tietojärjestelmissä tukevat rajapinnat projektin osapuolten esittämien tarpeiden pohjalta. Tällaisena tarpeena on tunnistettu OID-tunnuksia luovan palvelun rajapinta. Myös OID-rekisterin tarvetta on selvitetty, mutta siihen liittyviä konkreettisia käyttötarpeita ei ole noussut toistaiseksi esiin.

Tämä on OID-generaattorin rajapinnan määrittelevän WSDL-tiedoston ja sen hyödyntämisen kuvaava dokumentti. Määrittelyssä on hyödynnetty Sosiaali- ja terveydenhuollon OID-tunnuksen käyttö -ohjetta (Ensio 2004) sekä ISO OID-yksilöintitunnuksen soveltaminen julkishallinnossa -suosituksen luonnosta (JHS 2005).

Tämän dokumentin luvussa 2 määritellään OID-generaattorin rajapinnassa käytettävät toiminnot ja tiedot, niiden merkitys sekä toimintoihin liittyvät virhetilanteet, joiden pohjalta SerAPI:n teknisiä käytäntöjä (SerAPI 2005) noudattaen on luotu liitteessä 1 oleva tekninen WSDL-rajapintamäärittely. Luvussa 3 kuvataan kuinka generaattoria käytetään kahdessa yleisessä käyttötarkoituksessa: henkilön ja CDA-dokumentin yksilöintiin. Liitteessä 2 on kuvattu suomalaisen hyödyntämisohjeen mukaiset OID-solmuluokat.

Yhteentoimivien rajapintatoteutusten rakentamiseksi sovelluksiin dokumentin luvut 2 ja 3 sekä liite 1 ovat normatiivisia ja tarkasti noudatettavia.

SerAPI-hankkeessa WSDL-rajapintamäärittelyksen pohjalta tehty esimerkkiteotus OID-generaattorista ja sitä WSDL/SOAP-rajapinnan kautta käyttävästä palvelusta on saatavilla SerAPI-hankkeen kautta hankkeen osapuolille.

1.1 Dokumentissa käytetyt tietotekniset ja terveydenhuollon käsitteet

CDA	Clinical Document Architecture, terveydenhuollon sähköisten dokumenttien rakenteen ja sisällön määrittelyissä käytetty HL7-standardi. Ks. lisätietoja http://www.hl7.org/Library/standards_non1.htm#CDA
Jatkoalueen solmuluokka	Mikäli solmuluokalla on useampi kuin yksi generaattori, generaattorit voidaan määrittellä kaavalla $1000 * \text{solmuluokka} + n$, missä $n=1,2,3, \dots, 999$. (JHS 2005)
Juuri	OID-tunnus, jonka alle juuren haltija voi luoda oman tunnushierarkian.
Objekti	Esimerkiksi organisaatio, asiakirja, koodisto. Sähköiseen asiakirjaan viitattaessa objektilla tarkoitetaan asiakirjan rakenneosaa.
ISO OID	ISO Object Identifier. Objektin yksilöivä tunnus, joka perustuu ISO-standardiin ja jota käytetään digitaalisessa kommunikaatiossa mm. organisaatioiden, dokumenttien ja henkilöiden yksilöintiin.
Solmu	Solmulla tarkoitetaan tietorakenteissa kohtaa, jossa siihen liittyy alemman tason rakenteita. (JHS 2005)
Solmuluokka	Solmuluokalla tarkoitetaan ISO OID -järjestelmässä numerotunnusta, joka on annettu kullekin objektiluokalle ISO OID -järjestelmän eri hierarkiatasoilla. Esimerkiksi yrityksen ja yhteisön

	<p>tai toimipaikkatason juuren alle on annettu solmuluokan numerotunnukset 11 (asiakirjat), 12 (ohjelmistot), 13 (laitteet) jne. (JHS 2005).</p> <p>Suomessa käytettävän hyödyntämishojeen mukaiset solmuluokat on kuvattu liitteessä 2.</p>
--	--

2 Rajapinnan kuvaus

Tässä luvussa esitellään OID-generaattorin rajapinnan sisältämät operaatiot syötteineen ja paluuarvoineen. Lisäksi tuodaan esiin tehtyjen suunnitteluratkaisuiden ja rajauksien perustelut.

2.1 Vaatimukset ja rajaukset

OID-generaattoripalvelun on kyettävä luomaan uusia OID-tunnuksia palvelun hallinnoimien OID-juurien ja solmuluokkien alle hyödyntämisoheeseen (Ensio 2004) mukaisesti. Lisäksi palvelulta on voitava ohjelmallisesti selvittää, mitä juuria ja solmuluokkia se hallinnoi.

Rajapinnan ulkopuolelle on rajattu toimipaikkojen yksilöinti ja sähköisten asiakirjojen sisältämien objektien yksilöinti. OID-tunnuksien luonti toimipaikoille tässä dokumentissa kuvatun rajapintamäärittelyksen toteuttavaa OID-generaattoria käyttäen on kuitenkin teknisesti mahdollista, mutta tarvittavat metatiedot on syötettävä muilla keinoin.

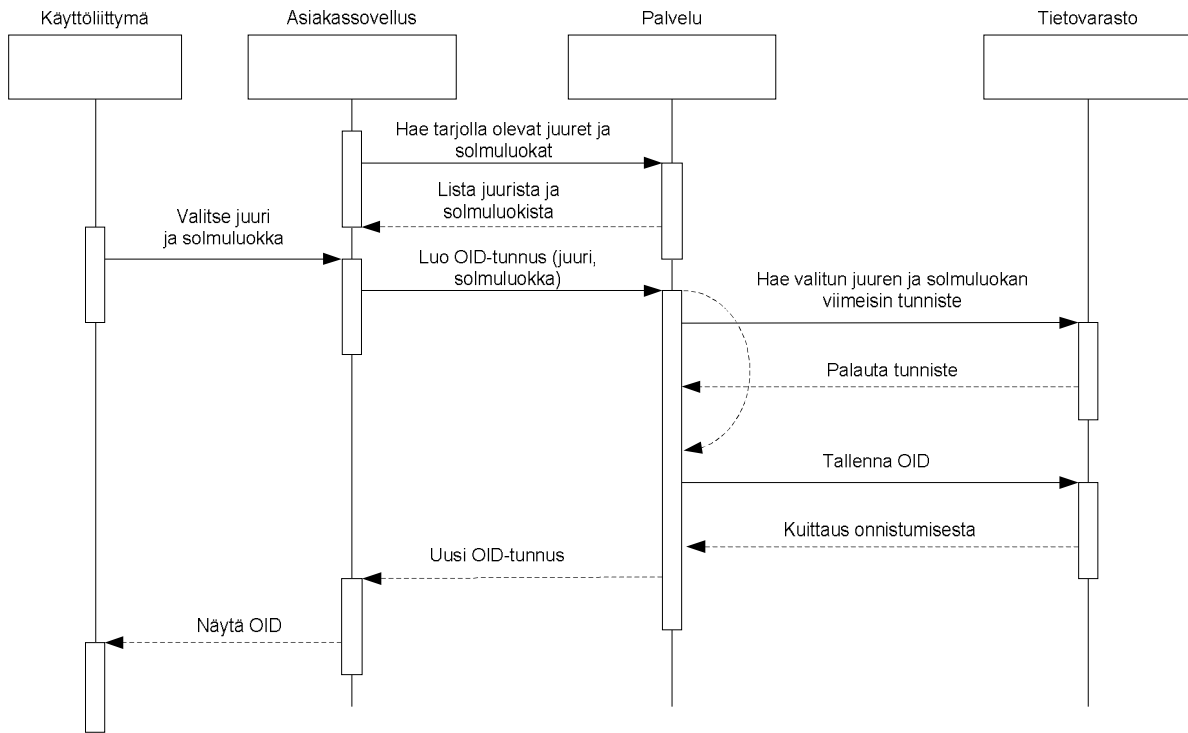
Tässä määrittelyssä oletetaan, että sähköisiä asiakirjoja tuottava sovellus tuottaa itse tarvittavat asiakirjan sisältämien objektien yksilöintitunnukset asiakirjalle annetun OID-tunnuksen alle. Ks. lisätietoja sähköisten asiakirjojen objektien yksilöinnistä (Ensio 2004), luku 4.3.

2.2 Rajapinnan toiminnallisuus

OID-generaattori luo ja palauttaa sovellukselle Stakesin OID-hyödyntämisoheeseen mukaisen OID-tunnuksen yksilöitävälle objektille. Palvelun on sisällettävä ainakin tieto seuraavasta vapaasta OID-tunnuksesta kussakin hyödyntämisoheeseen mukaisessa kategoriassa. Jos asiakirjan tunnuksen kuuluu vuosiluku, se luodaan palvelimella. Jos uuden tunnuksen tallentaminen tietovarastoon ei onnistu, tunnusta ei lähetetä asiakassovellukselle. Virhetilanteessa asiakassovellukselle palautuu virhesanoma, joka sisältää lyhyen kuvauksen virheestä. Tällainen on mm. tilanne, jossa uuden tunnuksen tallentaminen tietovarastoon ei onnistu, jolloin tunnusta ei myöskään lähetetä asiakassovellukselle.

Rajapinnan toteuttavan palvelun toiminnallisuudessa on tunnistettu kaksi erilaista tapaa. Ensimmäisessä tavassa palvelu hallinnoi juuria ja solmuluokkia. Tällöin vain palveluun määriteltyjen juurien ja solmuluokkien alle voidaan luoda OID-tunnuksia. Käytettävissä olevat juuret ja solmuluokat tulee saattaa asiakassovelluksen tietoon esim. paikallisen konfiguraation, erillisen hakemistopalvelun, tietojärjestelmän tai palvelulle tehtävän kyselyn avulla. Tässä dokumentissa kuvatussa rajapinnassa on määritelty operaatio palvelulle tehtävään juuret ja solmuluokat palauttavaan kyselyyn.

Toinen tapa on tehdä avoin OID-generaattori, joka on toteutettu niin, että käytettäviä juuria ja solmuluokkia ei ole rajattu. Tällöin vastuu käytössä olevista juurista ja solmuluokista siirtyy palvelua käyttävän asiakassovelluksen vastuulle. Tällöin vastuu käytettävistä olevista juurista ja solmuluokista on palvelua käyttävällä asiakassovelluksella, eli juurien ja solmuluokkien hallinta on asiakassovelluskohtaista.



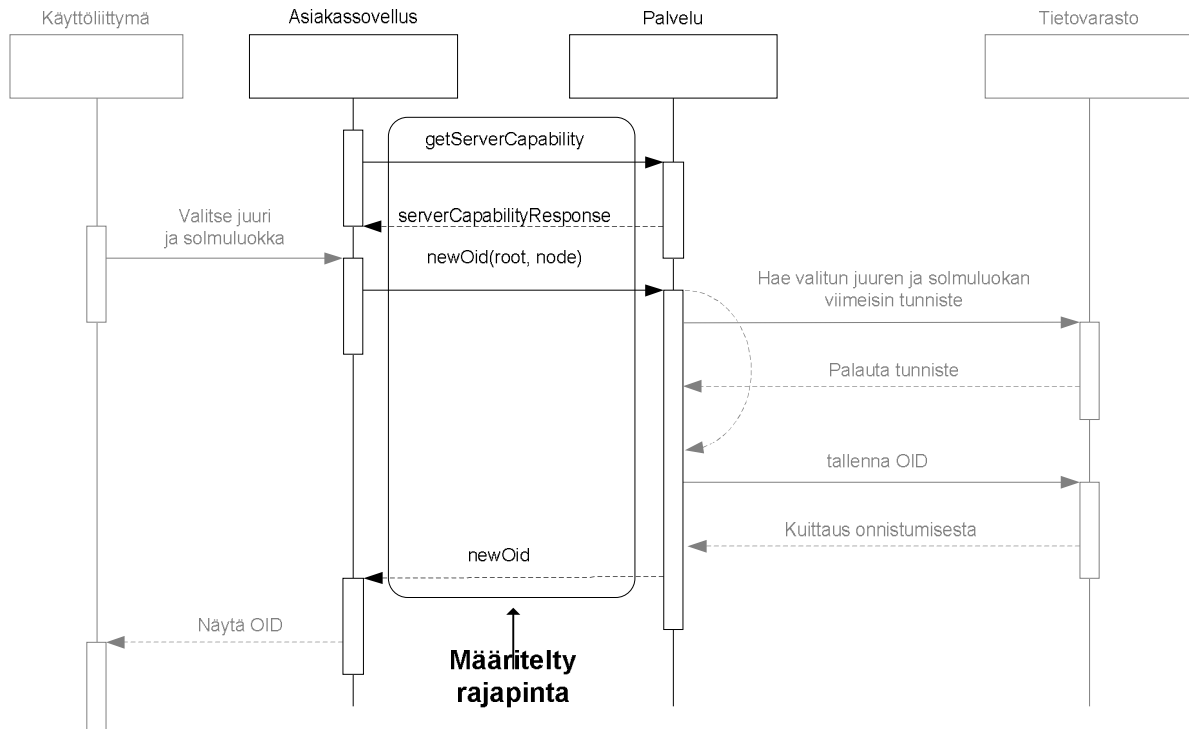
Kuva 1. OID-tunnuksen luonti

Kuvassa 1 on esimerkkinä onnistunut OID-tunnuksen luominen. Käyttöliittymälle toiminnallisuuden tarjoava asiakassovellus kutsuu OID-generaattorin rajapinnan toteuttavaa palvelua käyttäjän syöttämien valintojen mukaisilla parametreilla. Tässä esimerkissä palvelu hallinnoi käytettäviä juuria ja solmuluokkia. Kuva 2 esittää tässä dokumentissa kuvatun OID-generaattorin rajapinnan osuuden kuvan 1 esimerkistä WSDL-rajapintamäärittelyssä käytettävien operaatioiden ja parametrien nimiä käyttäen. Rajapinnan osuus on kuvattu kaaviossa tummemmalla sävyllä.

Kuvissa 1 ja 2 näkyvät kaksi asiakassovelluksen ja OID-generaattorin välistä operaatiota, jotka ovat osa määriteltyä rajapintaa. Operaatiot koostuvat rajapintatasolla kutsusta ja paluuarvo(i)sta. Ensimmäisessä operaatiossa (Hae tarjolla olevat juuret ja solmuluokat / *serverCapability*) asiakassovellus selvittää, mitä juuria ja solmuluokkia kutsuttava palvelu hallinnoi. Toisessa operaatiossa (Luo OID-tunnus / *newOid*) luodaan uusi tunnus halutun juuren ja solmuluokan alle. Palvelu hakee käyttämässään tietovarastosta edellisen ko. juuren ja solmuluokan alle luodun tunnuksen ja luo tämän tiedon pohjalta uuden tunnuksen. Luotuaan uuden tunnuksen, palvelu tallettaa sen tietovarastoon ja palauttaa tunnuksen asiakassovellukselle.

2.3 Rajapinnan operaatiot

Rajapinnalla on kolme operaatiota. Operaatiolla *newOid* voidaan luoda yksilöintitunnus solmuluokille 11-20 ja operaatiolla *newOidForPerson* solmuluokille 22-24 (solmuluokat on kuvattu liitteessä 2). Operaatio *serverCapability* hakee palvelun hallinnoimat juuret solmuluokkineen. Yksilöintitunnuksia voidaan luoda vain listatuille (tai palvelun muuten hallinnoimille) juurille ja solmuluokille. Solmuluokkien 21, 25 ja 26 OID-tunnukset tulevat suoraan Stakesin OID-juuren alle.



Kuva 2. Rajapinnan kuvaama osuus

2.3.1 newOid

Operaatio ottaa yhtenä yhdistettynä parametrina juuren ja solmuluokan, joiden alle halutaan luoda uusi yksilöintitunnus. Operaatio palauttaa uuden yksilöintitunnuksen ja sen luontiajan tai virheilmoituksen.

Syöteparametrit:

- juuri ja solmuluokka yhtenä merkkijonona (*rootAndExtension*), esimerkiksi "1.2.246.537.10.1714953.11" (uusi P-S shp potilasasiakirja). Tunnukseen mahdollisesti tulevaa vuosilukua tai versionumeroa ei ole syöteparametrissa.

Paluuarvot:

- uusi OID-tunnus annetun juuren ja solmuluokan alta (*newOid*), esimerkiksi "1.2.246.537.10.1714953.11.2005.313" (P-S shp potilasasiakirja 313 vuonna 2005).

Pakollisuus:

- newOID on toteutettava kaikissa palveluissa, jotka luovat tunnuksia solmuluokkien 11-20 alle. Kaikki parametrit ovat pakollisia.

2.3.2 newOidForPerson

Syöteparametrit:

- juuri ja solmuluokka yhtenä merkkijonona (*rootAndExtension*), esimerkiksi ”1.2.246.537.10.1714953.22” (uusi P-S shp tilapäinen henkilötunnus). Tunnukseen mahdollisesti tulevaa vuosilukua ei ole syöteparametrissa.
- henkilön sosiaaliturvatunnus (*ssn*), esim. ”241270-963R”

Paluuarvot:

- uusi OID-tunnus annetun juuren ja solmuluokan alta (*newOid*), esim. ”1.2.246.537.10.1714953.22.2004.1970122496328”.

Pakollisuus:

- *newOIDForPerson* on toteutettava kaikissa palveluissa, jotka luovat tunnuksia solmuluokkien 22-24 alle. Kaikki parametrit ovat pakollisia.

2.3.3 serverCapability

Syöteparametrit:

- ei syöteparametreja (XML-pohjaisessa rajapinnassa tyhjä merkkijono - ei varsinaisesti vaadi mitään syötettä)

Paluuarvot:

- lista generaattoripalvelun hallinnoimista juurista solmuluokkineen (*rootWithExtension*). Koska juuren lisäksi tarvitaan aina myös solmuluokka, palvelu ei palauta pelkkiä juuria vaan aina myös sen alla olevat palvelun hallinnoimat solmuluokat.

Pakollisuus:

- *serverCapability* ei ole pakollinen toteutettava kaikissa palveluissa. Ellei sitä toteuteta, palvelun hallinnoimat juuret ja solmuluokat on saatettava muulla tavalla palvelun käyttäjien tietoon. Kaikki parametrit ovat pakollisia.

2.4 Virhetilanteet

Palvelu voi asiakassovelluksen tekemän palvelupyynnön suorittamisen aikana ajautua virhetilanteisiin, jotka voivat johtua

- puutteellisista syötetiedoista (esim. puuttuva solmuluokka tai henkilötunnus)
- virheellisistä syötetiedoista (esim. juuri tai solmuluokka, jota palvelu ei hallinnoi)
- palvelun suoritusympäristössä tapahtuneista poikkeustilanteista

Virhetilanteessa palautetaan virheilmoitus (*exception*), joka sisältää virhekoodin ja selitteen/ilmoituksen. Tässä määriteltyjä virheitä ovat

- *GeneralFailure* Yleinen virhe
- *NotImplemented* Toimintoa ei ole toteutettu: (operaation nimi)
- *MissingParameter* Pakollinen tieto puuttuu: (parametrin nimi)

- InvalidParameter Virheellinen tieto (parametrin nimi)
- UnknownNode Väärä juuri tai solmuluokka

Jos palvelu ei pysty tallettamaan uuden tunnuksen tietoja, tunnusta ei lähetetä asiakassovellukselle. Tällä menettelyllä vältetään tilanne, jossa asiakas saa käyttöönsä tunnuksen, jota ei ole kirjattu minnekään.

Mikäli tunnuksen vieminen tietovarastoon onnistuu, mutta sen lähettäminen asiakkaalle epäonnistuu, asiakkaan kannalta tilanteessa ei ole tapahtunut vahinkoa (tulkitaan tunnuksen luonnin epäonnistumisena, voidaan esim. yrittää uutta tunnuksen luontia). Ei ole myöskään vaadittu, että kaikkia luotuja tunnuksia täytyisi käyttää numerojärjestyksessä, joten ei ole nähty tarpeelliseksi kehittää protokollaa luodun tunnuksen vastaanoton varmistamiseen.

Taulukko 1. Virhekoodit

Koodi	Ilmoitus esim.	Merkitys
GeneralFailure	Yleinen virhe	suoritusympäristön poikkeustilanne, erittelemätön virhe tms.
NotImplemented	Toimintoa ei ole toteutettu: (operaation nimi)	kutsuttavaa operaatiota ei ole toteutettu palvelussa
MissingParameter	Pakollinen tieto puuttuu: (parametrin nimi)	jokin pakollisista tai operaatioon vaadituista tiedoista (esim. juuri, solmuluokka tai henkilötunnus) puuttuu kutsun parametreista
InvalidParameter	Virheellinen tieto (parametrin nimi)	jokin saaduista syötteistä oli väärän muotoinen, esim. kirjaimia tai 0:lla alkavia OID-osia, henkilötunnus, jota ei voi muuntaa OID-muotoon tms.
UnknownNode	Väärä juuri tai solmuluokka	pyydetty tunnusta juuren tai solmuluokan alta, jota palvelu ei hallinnoi

3 OID-tunnuksen ja generaattorin käyttötapoja

Useimpien objektien yksilöinti noudattaa samaa kaavaa, eli halutun juuren alta valitaan sopiva solmuluokka ja sen alta haetaan seuraava vapaa arvo. Juuren, solmuluokan ja vapaan arvon (sekä mahdollisten muiden attribuuttien, kuten esim. vuosiluku) yhdistelmästä muodostuu uusi yksilöintitunnus. Yksilöintiä kaipaavien objektien joukosta voidaan erottaa muutamia poikkeavia tapauksia, joita seuraavissa kappaleissa on käsitelty hiukan tarkemmin muutamien käyttötapojen kautta.

3.1 Henkilön yksilöinti

Stakesin ohjeiden (Ensio 2004) mukaan vain Stakesin OID-juuren alle voidaan luoda yksilötunnuksia solmuluokkiin 21, 25 ja 26 (asiakas/potilastunniste henkilötunnus, SV-numero ja terveydenhuollon ammattihenkilöstö). Stakesin juuren alla siis sijaitsevat oikeiden henkilötunnusten perusteella luodut yksilöintitunnukset suoraan em. solmuluokissa.

Toimipaikat voivat luoda yksilötunnuksia solmuluokkiin 22, 23 ja 24 (tilapäinen asiakas/potilastunniste henkilötunnus, potilasnumero ja henkilönumero). Toisin sanoen toimipaikat voivat luoda yksilötunnuksia henkilöille vain tilapäisiä henkilötunnuksia käyttäen.

Esimerkki (suluissa generaattorin lisäämä osuus):

- Tilapäinen henkilötunnus: 1.2.246.537.10.1234567.22(.2005.tilht-tunnus)
- Potilasnumero: 1.2.246.537.10.1234567.23(.2005.potilasnumero)

3.2 Sähköisen asiakirjan ja sen osien yksilöinti

Jos asiakirjoille luodaan OID-tunnukset suoraan terveydenhuollon organisaation juuren alle, generaattorin avulla voidaan varmistua siitä, että eri sovellukset saavat omiin dokumentteihinsa yksikäsitteiset OID-tunnukset (solmuluokan 11 alle). Toisaalta organisaation asiakirjahaaraan voidaan luoda alahaara myös kullekin järjestelmälle, joka luo OID-tunnukset oman haaransa alle (tällöin solmuluokkana on esim. Musti 1: 11001). Myös tällöin voidaan käyttää generaattoria, mutta eri järjestelmien luomien dokumenttien tunnuksien yksikäsitteisyys saavutetaan muutenkin¹. Lisäsolmuluokkien määrittelystä on kerrottu suosituksessa (JHS 2005).

Esimerkki (suluissa generaattorin lisäämä osuus):

- Organisaatiokohtainen asiakirjatunnus: 1.2.246.537.10.1714953.11(.2005.113)
- Järjestelmäkohtainen asiakirjatunnus: 1.2.246.537.10.1714953.11003(.2005.49)
- Toinen järjestelmäkohtainen asiakirjatunnus (sis. versionumeron 1):
1.2.246.537.10.1714953.11004(.1.12)

¹ jos järjestelmä luo tunnuksen itse, solmuluokan alla voi yksilöintiin käyttää asiakirjoissa vuotta tai ”versiota”, generaattorirajapinnassa ei kuitenkaan välitetä tietoa kumpaa tapaa käytetään, oletuksena on vuosi

Itse dokumentin lisäksi myös dokumentin rakenneosia voidaan yksilöidä. Yksilöinti voidaan kuitenkin toteuttaa ilman dokumentteja luovan ohjelmiston ulkopuolista OID-generaattoria antamalla kullekin rakenneosaselle dokumentin yksilöintitunnuksen alla juokseva numero.

Ensio (2004) selittää sähköisten asiakirjojen yksilöinnin seuraavalla tavalla:

"Sähköinen asiakirja tallennetaan rakenteisena objektimuotona ISO:n ASN.1 mukaisesti (esim. CDA/XML- rakenne). Sähköisen asiakirjan objektit järjestetään aikajärjestykseen. Sähköisen asiakirjan sisällä on mahdollista yksilöidä asiakirjan objektit juoksevasti numeerisella objektitunnuksella. Objektien juoksevalla yksilöinnillä varmistetaan osaltaan asiakirja eheys. Objektin sisällä voidaan objektin osat tunnistaa juoksevalla sanojen numeroinnilla."

Esimerkki:

CDA-dokumentti on saanut OID-tunnuksen 1.2.246.537.10.1714953.11.2005.313 (juuri 1.2.246.537.10.1714953, solmuluokka 11, vuosi 2005 ja dokumentin numero 313). Dokumentin sisältämät objektit saavat juoksevan numeroinnin heti dokumentin tunnuksen juureen:

Objekti 1: OID 1.2.246.537.10.1714953.11.2005.313.1,
Objekti 2: OID 1.2.246.537.10.1714953.11.2005.313.2 jne.

Objektien sisältökin voidaan yksilöidä samalla tavalla:

Objekti 1, sana 4: OID 1.2.246.537.10.1714953.11.2005.313.1.4

Lähteet

Ensio, A. Sosiaali- ja terveydenhuollon OID-tunnuksen käyttö, versio 1.0. Stakes, 22.11.2004. Viitattu 4.10.2005.

<http://www.stakes.fi/koodisto/Sosiaali-%20ja%20terveydenhuollon%20OID-tunnuksen%20käyttö.doc>

Ensio, A. OID-tunnus, toimipaikkarekisteri ja koodistopalvelin. Stakes, 7.2.2005. Viitattu 11.10.2005. <http://www.oskenet.fi/uploads/ck4zhi9jnulc.ppt>

ISO OID-yksilöintitunnuksen soveltaminen julkishallinnossa. JHS, 24.10.2005. Viitattu 7.11.2005. [http://www.jhs-suositukset.fi/intermin/hankkeet/jhs/home.nsf/files/JHS_OID_versio_05_2005-10-24/\\$file/JHS_OID_versio_05_2005-10-24.pdf](http://www.jhs-suositukset.fi/intermin/hankkeet/jhs/home.nsf/files/JHS_OID_versio_05_2005-10-24/$file/JHS_OID_versio_05_2005-10-24.pdf)

Web-sovelluspalveluiden teknisiä suosituksia. SerAPI-hanke, 2.3.2005. <http://www.uku.fi/tike/his/serapi/yhteys/selvitykset/WS-suositukset.doc>

Liite 1: OID-generaattorin WSDL-määrittely

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="OIDv1"
  targetNamespace="urn:serapi:oid"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:serapi="urn:serapi:oid"
  xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      targetNamespace="urn:serapi:oid">
      <xsd:element name="newOid" type="serapi:Oid" />
      <xsd:element name="newOidForPersonRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="rootAndNodeClass"
              type="xsd:string" />
            <xsd:element name="ssn" type="xsd:string" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="newOidRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="rootAndNodeClass"
              type="xsd:string" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:complexType name="Oid">
        <xsd:sequence>
          <xsd:element name="root" type="xsd:string" />
          <xsd:element name="extension" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:element name="serverCapabilityResponse">
        <xsd:complexType>
          <xsd:sequence maxOccurs="unbounded" minOccurs="0">
            <xsd:element name="rootAndNodeClass" type="xsd:string" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="serverCapabilityRequest"
        type="xsd:string" />
      <xsd:element name="exception">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="errorCode" type="xsd:string" />
            <xsd:element name="explanation"
              type="xsd:string" minOccurs="0" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="newOidForPersonResponse">
    <part name="newOidForPersonResponse" element="serapi:newOid" />
  </message>
  <message name="newOidForPersonRequest">
    <part name="newOidForPersonRequest"

```

```

        element="serapi:newOidForPersonRequest" />
</message>
<message name="newOidResponse">
    <part name="newOidResponse" element="serapi:newOid" />
</message>
<message name="newOidRequest">
    <part name="newOidRequest" element="serapi:newOidRequest" />
</message>
<message name="serverCapabilityResponse">
    <part name="serverCapabilityResponse"
        element="serapi:serverCapabilityResponse" />
</message>
<message name="serverCapabilityRequest">
    <part name="serverCapabilityRequest"
        element="serapi:serverCapabilityRequest" />
</message>
<message name="exceptionMessage">
    <part name="fault" element="serapi:exception" />
</message>

<portType name="Oid">
    <operation name="newOidForPerson">
        <input message="serapi:newOidForPersonRequest" />
        <output message="serapi:newOidForPersonResponse" />
        <fault name="exception" message="serapi:exceptionMessage" />
    </operation>
    <operation name="newOid">
        <input message="serapi:newOidRequest" />
        <output message="serapi:newOidResponse" />
        <fault name="exception" message="serapi:exceptionMessage" />
    </operation>
    <operation name="serverCapability">
        <input message="serapi:serverCapabilityRequest" />
        <output message="serapi:serverCapabilityResponse" />
        <fault name="exception" message="serapi:exceptionMessage" />
    </operation>
</portType>

<binding name="OidSOAPBinding" type="serapi:Oid">
    <wsdlsoap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="newOidForPerson">
        <wsdlsoap:operation
            soapAction="urn:serapi:oid/newOidForPerson" />
        <input>
            <wsdlsoap:body use="literal" />
        </input>
        <output>
            <wsdlsoap:body use="literal" />
        </output>
        <fault name="exception">
            <wsdlsoap:fault name="exception" use="literal" />
        </fault>
    </operation>
    <operation name="newOid">
        <wsdlsoap:operation soapAction="urn:serapi:oid/newOid" />
        <input>
            <wsdlsoap:body use="literal" />
        </input>
        <output>
            <wsdlsoap:body use="literal" />
        </output>
    </operation>
</binding>

```

```
</output>
<fault name="exception">
  <wsdlsoap:fault name="exception" use="literal" />
</fault>
</operation>
<operation name="serverCapability">
  <wsdlsoap:operation
    soapAction="urn:serapi:oid/serverCapability" />
  <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="OidService">
  <port name="OidServiceSOAPPport"
    binding="serapi:OidSOAPBinding">
    <wsdlsoap:address
      location="http://localhost/OidService/services/OidSOAPPport" />
    </port>
  </service>
</definitions>
```

Liite 2. Suomalaisen hyödyntämisohjelman mukaiset OID-solmuluokat

(Ensio 2004)

solmu- luokka (xx)	jatkoalueen solmuluokat (xx)	selite
5		teknilliset koodistot, nimikkeistöt ja luokitukset
6		koodistot, nimikkeistöt ja luokitukset
10		palveluntuottaja ja toimipaikka
11	11 001-11 999	asiakirjat
12	12 001-12 999	ohjelmistot
13	13 001-13 999	laitteet
14	14 001-14 999	hoito- ja palveluketjut
15	15 001-15 999	hoitojonot
16	16 001-16 999	laskutuksen (sis. maksusitoumukset)
17	17 001-17 999	logistiikan tunnus (sis. kuljetustunnus)
18	18 001-18 999	sanomaliikenteen osapuoli
19	19 001-19 999	rekisterinpitäjä/toimintayksikkö
20	20 001-20 999	näytetunniste
21		asiakas/potilastunniste henkilötunnus
22	22 001-22 999	tilapäinen asiakas/potilastunniste henkilötunnus
23	23 001-23 999	potilasnumero
24	24 001-24 999	henkilönumero
25		SV-numero
26		terveydenhuollon ammattihenkilöstö
27	27 001-27 999	roolin yksilöinti