

Datan luonti ja siirto DICOM-standardilla

Jukka Pollari

Tampereen yliopisto
Informaatiotieteiden yksikkö
Tietojenkäsittelyoppi
Pro gradu -tutkielma
Ohjaaja: Jyrki Nummenmaa
Huhtikuu 2012

Tampereen yliopisto

Informaatiotieteiden yksikkö

Tietojenkäsittelyoppi

POLLARI, JUKKA: Datan luonti ja siirto DICOM-standardilla

Pro gradu-tutkielma, 56 sivua, 4 liitesivua

Huhtikuu 2012

Tässä tutkielmassa kuvataan, kuinka DICOM-standardilla luodaan dataolioita ja millä tavalla niitä liikutetaan eri ohjelmistojen välillä. Tutkielmassa kuvataan DICOM-dataolion luonti ylätasolta tosielämän kuvauksista kohti varsinaista haarautuvaa rakenteista dataa. Tutkielmassa havainnollistetaan DICOM-standardin tarjoamat työkalut datan siirtoon. Datan siirtoon käytettyjä palvelukutsuja havainnollistetaan FSP-prosessikaavioiden avulla. Tutkielmassa kuvataan standardin yleisemmät ongelmat ja yritetään etsiä niihin ratkaisuja. Ongelmien ratkaisuksi esitetään pääasiassa XML-kieleen perustuvia datan esitysmuotoja ja tiedonsiirtomenetelmiä. Lisäksi pohditaan, mitkä voisivat olla DICOM-standardin tulevaisuuden suuntaukset.

Avainsanat ja -sanonnat: DICOM, XML, Web Service, SOAP, integraatiot, kuva, sairaala, tiedonsiirto, tietoliikenne, IOD, SOP, DIMSE, FSP, LTS, JPEG 2000.

Alkusanat

Olen ollut vuodesta 2008 lähtien töissä MediWare Oy -yrityksessä. Haluan kiittää MediWare Oy -yritystä joustavuudesta ja opintovapaan antamisesta sekä muissa opiskeluissa, että tämän tutkielman tekemisessä. Erityisesti MediWare Oy -yhtiöstä haluan kiittää Mika Niemelää ja Päivi Leppälampea opiskelujen sallimisesta töiden ohessa. Vuonna 2008 olin MediWare Oy -yhtiössä mukana toteuttamassa DICOM-integraatioprojektia sikiöseulonnan tarpeisiin. DICOM-projektissa työskentelytoverinani oli todellinen Caché-guru Jyrki Tuomi, jonka ammattitaitoa en voi lakata ihailmasta. Mediware Oy -yhtiön projektissa toteutettiin kaksi eri DICOM-palveluluokkaa, joista molemmista kerrotaan tässä tutkielmassa.

Haluan kiittää vaimoani Mariaa ymmärtämisestä sekä työn, opiskelun ja lapsenhoidon aikataulujen yhteensovittamisesta. Tämän tutkielman kirjoittamisen aikana tyttäremme Liisa kasvoi kaksivuotiaaksi. Ilman häntä arkinen elämämme olisi ollut kovin tylsää. Haluan kiittää myös omia vanhempiani gradun tekemisen kannustamisessa. Lisäksi kiitän myös vaimoni vanhempia, jotka auttoivat kirjoitustilojen hankkimisessa.

Erittäin iso kiitos tästä pro gradu -tutkielmasta kuuluu Tampereen yliopiston tietojenkäsittelyopin professorille Jyrki Nummenmaalle. Hän antoi opastusta siihen, mistä näkökulmasta tutkielman aihepiiriä kannattaa lähestyä. Lisäksi hän auttoi tutkielman tekemisen aikana varsinkin siinä, mihin tutkielmassa kannattaisi eniten keskittyä. Kiitos myös Tampereen yliopiston Pauli Borodulinille, joka auttoi LTSA-ohjelmaan liittyvän Java-ongelman kanssa.

Tampereella 9.4.2012 Jukka Pollari

Sanasto

CRC - Cyclic redundancy check, tarkisteavain, jolla voidaan tutkia datassa ilmenevät virheet. CRC-avaimen käyttö on yleistynyt voimakkaasti Internetin verkkoliikenteen kasvaessa.

RIS - Radiology Information system. Radiologia-osaston tietojärjestelmä

HIS - Hospital Information system. Sairaalan tietojärjestelmä

SOP - Service-Object Pair. DICOM-standardissa tiedon (IOD) ja palvelun yhteen kytkevä kokonaisuus

IOD - Information Object Definition. VR-tietotyyppien avulla koodattu DICOM-dataolio, joka sisältää tutkimukseen liittyvän datan, esimerkiksi potilaan tunnistetiedot ja kuvadatan.

Modaliteetti - DICOM-verkkoon liitetty laite, jolla kuvataan ja tallennetaan lääketieteellisiä kuvia. Modaliteetti voi olla esimerkiksi röntgen- tai ultraäänilaite, mutta myös esimerkiksi digitaalikamera. Tässä tutkielmassa modaliteetista käytetään myös nimeä kuvantamislaitte.

Application Entity - DICOM-verkossa sijaitseva ohjelmisto tai muu kohde. AE Title, eli ohjelmiston nimi konfiguroidaan usein ohjelmiston vastapäin asetuksiin.

TCP/IP - Internetissä yleistynyt verkkoliikenneprotokolla, jota DICOM käyttää oman siirtoprotokollansa alapuolella

HL7 - Health Level 7. Lääketieteessä käytettäviä standardeja julkaiseva kansainvälinen yhdistys, jolla on joka maassa oma alijärjestö. Uusin DICOM-standardi sisältää osittaisen DICOM-datan muunnoksen HL7-standardin muotoiseksi dataksi. HL7 versiota 2 noudattavat sanomat ovat suosittuja suomalaisten lääketieteen sovellusten tiedonsiirrossa.

WSDL - Web Services Description Language. W3C-järjestön julkaisema XML-muotoinen kuvaus Web Service -tyyppisille kutsuille.

ISO - International Standards Organization. Maailmanlaajuinen eri alojen standardeja julkaiseva järjestö

PACS - Picture archiving and Communication System. Sairaalan kuva-arkisto, johon tallennetaan ja josta voidaan tutkia eri ohjelmistoista talletettuja tutkimuksia.

3G - Third generation. Yleinen nimitys NMT ja GSM-teknologioiden jälkeiselle matkapuhelinteknologialle eli "kolmannelle sukupolvelle".

Sisällys

Alkusanat	iii
Sanasto.....	iv
1. Johdanto.....	1
2. Standardin historia ja käyttötarkoitus	3
2.1. Standardin historia	3
2.2. Standardin käyttötarkoitus	4
2.3. Standardin sisältö ja vuoden 2011 osiot	4
2.4. Tutkielman raja- ja standardin näkökulmasta.....	4
3. DICOM-dataoliot ja datan rakenne	6
3.1 Olioiden rakenne	6
3.2 Information Object Definition.....	6
3.3 VR-tietotyypit.....	8
3.4 SQ VR, sisäkkäiset dataoliot	10
3.5 Sanakirja ja tagiryhmät	11
3.6 Dataelementit	12
3.7 Unique Identifiers (UID).....	13
3.8 Dataelementit IOD-olioissa	13
3.9 Binaaridata.....	13
3.10 Kuvadata.....	14
3.11 Kuvanpakkaus	15
4. DICOM-tietoliikenne	17
4.1 DICOM-verkkoliikenteen hierarkia.....	17
4.2 SOP-luokka.....	18
4.3 Yksilöinti DICOM-verkossa.....	20
4.4 Palvelujen käyttöroolit.....	20
4.5 DIMSE-palvelut	21
4.6 Upper Layer -protokolla (DICOM Associations).....	22
4.7 Yhteysneuvottelu (Association Negotiation).....	23
4.8 Datat siirto (P-Data-TF) ja yhteyden katkaisu.....	26
4.9 Tutkielmassa käytetyt prosessikaaviot (FSP)	26
4.10 FSP-kaavio DICOM UL -protokollasta.....	27
4.11 SOP-luokat ja eri palvelutyypit	29
5. Standardin ongelmat.....	35
5.1 Point-to-point -tietoliikenne.....	35
5.2 Tietoliikenteen rajoittaminen DICOM-verkossa.....	36
5.3 Binaaritavuihin perustuva koodauksen edut ja haitat.....	37
5.4 Monimutkaisia tietotyyppisiä, määrittelyitä ja elementtejä.....	37

5.5 Parillisuussääntö.....	38
5.6 DICOM-datan muunnostarpeet	38
6. Vaihtoehtoisia lähestymistapoja	40
6.1 DICOM datan ja XML-tiedon rakenne	40
6.2 XML-muunnokset ja vuoden 2011 HL7-XML -muunnos	41
6.3 Datan siirto HTTP-protokollan avulla.....	41
6.4 Kuvaus Web Service -palveluista ja käytöstä DICOM-datan välityksessä...	42
6.5 HTTP-protokolla ja transaktioiden ongelmat	43
6.6 WADO.....	44
6.7 DIM Web Service	45
6.8 Application Hosting.....	46
7. Standardin tulevaisuus.....	49
7.1 Kuvantamisen tarpeet tulevaisuudessa	49
7.2 DICOM ja mobiililaitteet	50
7.3 DICOM-verkkojen tulevaisuus terveyslaitoksissa	51
7.4 Kansallinen terveysarkisto	51
7.5 DICOM-standardin kehityssuunta	52
8. Yhteenveto.....	53
Viiteluettelo	54
Liitteet.....	57

1. Johdanto

Tässä tutkielmassa syvennyttään DICOM (Digital Communications in Medicine) -standardiin. DICOM-standardia käytetään yleisesti datan siirtoon erilaisten lääketieteellisten ohjelmistojen välisessä keskustelussa. Standardi on suunniteltu toimimaan tietoverkossa TCP/IP-protokollan päällä. DICOM-protokolla sisältää kaksi tasoa. SOP (Service Object Pair) kokoaa yhteen tietoliikennevälineen sekä dataolion. SOP sisältää päällä toimivan erilaisia tietoliikennepalveluita tarjoavan ryhmän nimeltään DIMSE (DICOM Message Service Elements) sekä IOD (Information Object Definition) -dataolion. Varsinaista dataa liikutetaan TCP/IP-protokollan päällä ylemmän tason (Uppel Layer) verkkoliikenneprotokollan avulla.

Tutkielman tutkimusongelma on DICOM-standardin ongelmat ja tulevaisuuden näkymät Internetin avoimessa verkkoliikenteessä. DICOM perustui alun perin omaan point-to-point -protokollaan, mutta nykyään data välitetään TCP/IP-protokollan avulla. Tietoliikennearkkitehtuuri kuitenkin perustuu edelleen point-to-point -tyyppiseen kahden kohteen väliseen tiedonsiirtoon. Tämä tuo mukanaan tiettyjä rajoituksia. Tutkielmassa pohditaan, soveltuuko DICOM sellaisenaan tietoliikennestandardiksi Internetissä. Avoimemman datan ongelmiin on olemassa erilaisia XML-pohjaisia ratkaisuja myös standardin puolesta. Tutkielmassa tuodaan esille näitä ratkaisuja ja paneudutaan myös niiden ongelmiin. Lisäksi tutkielmassa pohditaan, mitkä ovat tulevaisuuden haasteet ja kehityssuunta sairaalaympäristön näkökulmasta. Tutkielmassa tehdään muun muassa kirjallisuuskatsaus tuoreimpiin artikkeleihin sekä muuhun DICOM-kirjallisuuteen. Itse standardiin perehdytään DICOM-standardin dokumentaation kautta.

Tutkielmassa kuvataan aluksi DICOM-standardin käyttötarkoitus ja syyt, jotka johtivat sen kehittämiseen. Itse standardin sisällöstä tässä tutkielmassa keskitytään datan muodostamiseen, kuvadataan ja datan siirtoon. Datat siirron havainnollistamiseksi tutkielman liitteeksi tehdään erillinen FSP-prosessikaaviokuvaus. Standardin palveluluokkien esittelyssä ja Upper Layer -protokollan kuvauksissa esitellään myös siihen liittyvä FSP-prosessikaavio. FSP-kaavion lisäksi datan siirtoa havainnollistetaan Labeled Transition System (LTS) -tyyppisten graafisten esitysten avulla. Tarkoitus on yrittää antaa mahdollisimman selkeä kuva siitä, mitä dataa tarkalleen siirretään missäkin tilanteessa. Tämän tutkielman ohessa toteutettu FSP-kuvaus löytyy kokonaisuudessaan tämän tutkimuksen liitteestä 1. Lisäksi liitteessä 2 on LTSA-ohjelmalla suoritettu käännöstulos liitteen 1 FSP-kuvauksesta.

Tässä tutkielmassa ei käsitellä aivan kaikkia DICOM-standardin osioita. Tutkielmassa keskitytään pääasiassa niihin standardin osioihin, jotka koskevat tiedon luontia ja tiedonsiirtoa DICOM-verkossa. Myös kuvadataan muodostamisen kuvaaminen on tärkeitä, koska DICOM-standardia käytetään pääasiassa kuvien siirtoon. Tutkielmassa kuvataan DICOMin oma BMP-kuvan rakenne. Lisäksi kuvanpakkausmenetelmistä kuvataan JPEG 2000 -pakkauksen ominaisuudet ja haasteet DICOM-verkossa.

Tutkielma etenee DICOM-standardin kuvauksen jälkeen erilaisiin ongelmiin DICOM-datassa ja sen jälkeen esittää ratkaisuja näihin ongelmiin. Ratkaisuja löytyy esimerkiksi XML-kieleen perustuvilla datan esitysmalleilla. Lisäksi Web Service -ratkaisuilla voidaan korvata standardin tiedonsiirtoon liittyviä ongelmia. Web Service -ratkaisuja on tarjolla sekä standardin, että ulkopuolisten tahojen toimesta. Lopuksi tutkielmassa pohditaan standardin tulevaisuuden suuntauksia.

Toisessa kappaleessa kuvataan standardin kehityshistoria ja esitellään kuinka standardia käytetään yleisimmissä tilanteissa sairaalan verkossa. Kappaleessa kuvataan myös standardin kehitystä vuoden 2011 versioon. Lisäksi kappaleessa tarkennetaan vielä niitä standardin osia, jotka jätetään käsittelemättä tässä tutkielmassa.

Kolmas kappale keskittyy DICOM VR (Value Representation) -muotoisen datan muodostamiseen. Ensimmäisenä kuvataan DICOM-dataolio, eli esitellään tosielämän olioiden mallinnus VR-dataksi. Sen jälkeen kuvataan binaaridataan perustuva elementtirakenne, tietotyypit ja tagikirjasto. Viimeiseksi kappaleessa kuvataan, kuinka kuvadata muodostetaan ja kuinka sitä voidaan pakata erilaisten algoritmien avulla.

Neljännessä kappaleessa keskitytään kuvaamaan DICOM VR -muotoisen datan välittämistä eri ohjelmistojen välillä. Ensin kuvataan ylemmän tason erilaiset tiedonsiirtopalveluluokat sekä niiden yhdistäminen aikaisemmin kuvattuun VR-muotoiseen dataan. Kappaleessa käsitellään myös varsinaista tiedonsiirtoa DICOM UL -protokollan avulla kahden laitteen välillä. Neljännen kappaleen lopussa kuvataan tärkeimmät SOP-luokat ja niiden käyttötarkoitukset. Sekä DICOM UL -protokollan tiedonsiirtoa, että SOP-luokkia yritetään havainnollistaa liitteen 1 FSP-prosessikaavion ja siitä koottujen LTS-diagrammien avulla.

Viidennessä kappaleessa perehdytään DICOM-standardin suurimpiin ongelmiin. Kappaleessa kuvataan ensin point-to-point -tyyppisen tietoliikenteen ongelmat. Seuraavaksi kuvataan DICOM VR -muotoisen datan vaikeaselkoisen binaarikoodauksen ongelmat.

Kuudennessa kappaleessa kuvataan point-to-point -ongelmaan sekä DICOM VR -muotoiseen datan vaikeaselkoisuuteen kehitetyt ratkaisut avoimemman datan näkökulmasta. Kappaleessa kuvataan XML-kieli lyhyesti ja se, kuinka DICOM on muunnettavissa XML-muotoon. Lisäksi kappaleessa perehdytään erilaisiin HTTP-protokollan avulla välitettäviin kyselyihin, kuten Web Service -ratkaisuihin. Kappaleessa perehdytään myös standardin vuoden 2011 version uudistuksiin ja käsitellään standardiuudistusten ongelmat. Lisäksi tehdään katsaus muihin mahdollisiin ratkaisuihin.

Seitsemännessä kappaleessa pohditaan standardin tulevaisuutta. Kappaleessa kuvataan nykyisten sairaalaympäristöjen DICOM-verkkojen tilanne verrattuna prosesseihin. Kappaleessa pohditaan, kuinka tilannetta voitaisiin parantaa. Lisäksi pohditaan langattoman verkon tarjoamia tulevaisuuden haasteita.

2. Standardin historia ja käyttötarkoitus

Tässä kappaleessa kuvataan aluksi standardin historia lyhyesti, jotta saadaan kuvaus siitä, mihin tarpeisiin se on alun perin kehitetty. Tämän jälkeen kuvataan standardin käyttötarkoitus nykypäivänä sairaalaverkkojen sisällä. Kappaleessa kerrotaan myös itse standardin eri osioista ja siitä, mihin osiin niistä tämä tutkielma keskittyy.

2.1. Standardin historia

Tietokonekerroskuvauslaitteet alkoivat yleistyä 1970-luvulla. Laitteiden alkuaikoina eri laitevalmistajilla oli erilaiset tavat välittää tietoa kuvauslaitteilta toisiin järjestelmiin. Koska tietojen välittämisessä ei ollut standardia tapaa, Yhdysvaltojen elektroniikkavalmistajien liitto National Electrical Manufacturers Association (NEMA) ja Yhdysvaltojen radiologien liitto American College of Radiology (ACR) kokivat tarpeelliseksi laatia standardit eri laitevalmistajien kuvaformaatteihin. Lisäksi koettiin tarpeelliseksi standardoida tiedonsiirron ja erilaisten kuviin liittyvien tutkimustietokantojen luonti. Vuonna 1983 perustettiin komitea kehittämään standardia näihin tarpeisiin. Järjestöjen tuottama ensimmäinen versio 1.0 ilmestyi vuonna 1985 [NEMA, PS 3.1, 2011] ja se oli nimeltään ACR-NEMA. ACR-NEMA käsitti seuraavat osiot:

- Point-to-point -protokolla
- VR-Datan muodostaminen
- Datan siirtoon käytettävät palveluluokat
- Sanakirja

[Jumppanen, Jaatinen, Toivanen 2005]

Toinen versio ilmestyi vuonna 1988 ja siihen oli lisätty laitemäärittelyt, ohjelmistoprotokollat sekä standardi data-sanakirja. Vuonna 1992 ilmestyi DICOM 3.0, joka sisälsi 13 osaa. Version 3.0 jälkeen standardin versioinnissa on alettu käyttämään vuosilukuja. Vuoden 2011 versio sisältää 20 osaa. Standardin uudet versiot sisältävät taaksepäin yhteensopivuuden DICOM 3.0-version kanssa. Tästä saatu hyöty on, että ohjelmiston tukiessa DICOM versiota 3.0, se on myös yhteensopiva 2011 vuoden version kanssa.

Nykyään DICOM on kaikkialla maailmassa yleistynyt tapa välittää kuvantamisdataa eri ohjelmistojen välillä. Esimerkkejä laajasta käytöstä on useita. DICOM on esimerkiksi standardoitu kansainvälisen ISO-standardin mukaisesti vuonna 2006 hyväksytyssä osiossa ISO 12052:2006 [ISO, 2006]. DICOM on tuettu laajasti myös Suomen lääketieteellisten järjestelmien kuvadatan siirrossa. Laajamittaisesta käytöstä kertoo esimerkiksi se, että DICOM-standardia tullaan käyttämään Suomessa Kansallisen terveystietokannan kuvien tiedonvälityksessä [Sosiaali- ja terveysministeriö, 2006]. Kansallisesta terveystietokannasta kerrotaan lisää kappaleessa 7.4.

2.2. Standardin käyttötarkoitus

Yleisimmin DICOM-standardia käytetään datan siirtämiseen erilaisten ohjelmistojen välillä sairaalan verkon sisällä. Erilaisia kuvatyyppejä ovat esimerkiksi CT (Computerized tomography, tomografiakuvat), MRI (Magnetic Resonance Imaging, magneettikuvaukset) ja ultraäänikuvat. [Jumppanen, Jaatinen, Toivanen 2005]. DICOM-dataolio sisältää kuvan lisäksi tutkittavan potilaan tiedot ja muita tutkimukseen liittyviä tietoja. Lisäksi tärkeässä roolissa on palveluolio, joka kuvaa sen, mitä datalla on tarkoitus tehdä.

Standardin avulla voidaan siirtää myös muutakin kuin kuviin liittyvää dataa. Standardin Structured Reporting -lisäosa tarjoaa mahdollisuuden kaikenlaisen potilasdatan välittämiseen sisäkkäisten dataelementtien avulla. Structured Reporting mahdollistaa käytännössä kaikenmuotoisen datan siirron. Sen avulla standardia voidaan esimerkiksi käyttää siirtämään sanelukstiä aaltoäänimuodossa [Kaldoudi, Karaiskakis, 2006]. Structured Reporting -lisäosaa käsitellään laajemmin kappaleessa 4.11.

ARC-NEMA suunniteltiin ennen TCP/IP -protokollan yleistymistä. Alkuperäisessä ARC-NEMA -standardissa oli mukana oma point-to-point -protokolla, mutta uusimmasta versiosta se on jätetty kokonaan pois. Point-to-point -arkkitehtuuri on kuitenkin jäänyt suurilta osin elämään myös TCP/IP-verkon päälläkin toimivassa DICOM-verkossa, vaikka varsinaisesta omasta alemman tason protokollasta on luovuttu. Tähän aiheeseen palataan kappaleessa 4, joka käsittelee tiedonsiirtoa. Lisäksi kappaleessa 6 kuvataan XML-muunnosta ja vaihtoehtoisia tapoja datan siirtämiseen.

2.3. Standardin sisältö ja vuoden 2011 osiot

Ensi silmäyksellä DICOM-standardi voi vaikuttaa laajalta, koska se sisältää useita tuhansia sivuja tekstiä. Suurin osa standardista on kuitenkin taulukoita ja sanakirjoja. Standardin tulkintaa helpottaa myös se seikka, että ensimmäisessä osiossa NEMA PS 3.1 [2011] annetaan sisällysluettelo myös muiden osioiden sisällöstä. Kyseiseen osioon perehtymällä saa hyvin kuvan siitä, mitkä ovat tärkeimpiä osioita omassa projektissa. Jokainen DICOM-ohjelmisto valitsee itse toteutettavat palvelut, joka myös vähentää osaltaan standardin tulkintaa. Joitakin osioita voi siis huoletta jättää lukematta, jos niitä ei toteuteta.

Standardista julkaistaan uusi versio nykyään 1-2 vuoden välein. Viimeisin versio on vuodelta 2011 ja se sisältää 20 osaa. Uusimpaan standardiversioon on lisätty kaksi kokonaan uutta osiota. Uusien kappaleiden lisääminen standardiin on kohtalaisen harvinaista. Pääasiassa uudistukset ovat vanhojen osioiden päivittämistä, esimerkiksi uuden lääketieteellisen tutkimustiedon lisäämistä tauluihin ja aikaisempien virheiden korjaamista.

2.4. Tutkielman rajaus standardin näkökulmasta

Tämä tutkielma keskittyy standardin niihin osiin, jotka ovat oleellisia dataolion muodostamisessa ja tietoliikennepalveluissa. Lisäksi tutkielmassa käsitellään standardin kuvadatan muodostus ja standardiin tehdyt XML-tyyppiset lisäykset.

Tutkielmasta pois jätettäviä osioita ovat esimerkiksi DICOM-muotoisiin tiedostoihin liittyvät asiat. Näitä ovat esimerkiksi DICOMDIR-tietokannat sekä DICOM-muotoisen datan tallennus medialaitteisiin.

Standardin tietoturva-asiat ovat yksi tutkielmasta pois rajattava osio. Tietoturva määritellään standardin puolesta lähinnä tiedostojen siirron tarpeisiin, eikä tässä tutkielmassa perehdytä DICOM-tiedostoihin. DICOM-tietoliikenteessä sairaalan sisäisen verkon tietoturva kuuluu yleensä verkosta vastaaville tahoille. Yksi yleinen ratkaisu on luoda VPN-verkko, jonka sisälle pääsevät vain luotettavat lähteet. Web Service -tyyppisten ratkaisujen tietoturva-asiat avoimessa verkossa ovat kuitenkin tämän tutkielman piirissä. Niistä kerrotaan kappaleessa 6.

Tässä kappaleessa kuvattiin standardin historia ja käyttötarkoitus. Lisäksi kuvailtiin sitä, millainen itse standardin dokumentaatio on. Seuraavassa kappaleessa lähdetään kuvaamaan sitä, kuinka DICOM-oliot mallinnetaan tosimaailman olioista. Lisäksi kuvataan VR-muotoisten dataelementtien luominen ja binaarimuotoisen datan haasteet. Lopuksi kerrotaan, kuinka DICOM-kuvadata muodostetaan.

3. DICOM-dataoliot ja datan rakenne

Tässä kappaleessa kuvataan DICOM-datan muodostaminen ylimmältä tasolta lähtien. Aluksi kuvataan se, kuinka tosielämään liittyvät potilastutkimukset mallinnetaan DICOM-olioiksi ja kuinka olion ominaisuudet muodostetaan. Tämän kappaleen tarkoitus on antaa kuva siitä, mikä DICOM-dataolio on ja millaisia ominaisuuksia sillä voi olla. Myöhemmissä kappaleen osioissa perehdytään moduuleiden yksittäisiin ominaisuuksiin. Osioissa esitellään se, kuinka ominaisuudet tarkalleen esitetään binaarimuotoisina dataelementteinä. Osioissa esitellään DICOM-standardin erityinen elementtien binaarimuotoinen VR-koodaus ja kirjastot, joita käytetään kaikkien dataelementtien esittämiseen. Ensin kuvataan tietotyypit, sitten DICOM-sanakirja. Sen jälkeen kuvataan, kuinka näiden avulla data kootaan varsinaisiksi dataelementeiksi. Myöhemmin kuvataan, kuinka dataelementit sulautuvat edellisessä kappaleessa kuvattuun DICOM IOD -olioon. Viimeisissä osioissa kuvataan kuvadatan muodostaminen ja se, kuinka kuvanpakkausalgoritmeja on mahdollista käyttää tiedonsiirron apuna. Tämä kappale on koottu pääasiassa DICOM-standardin dokumentaatiosta, mutta myös muita lähteitä on käytetty, joista mainitaan erikseen.

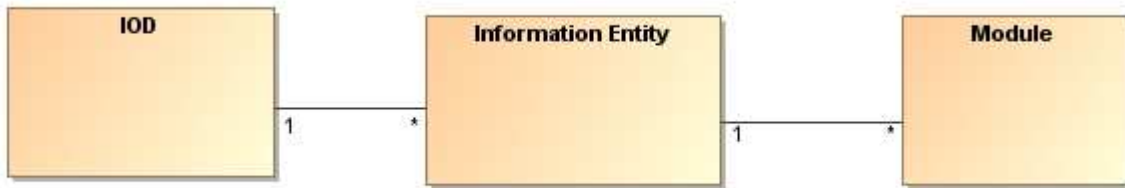
3.1 Olioiden rakenne

DICOM-standardi perustuu tosielämän tutkimusten mallintamiseen dataolioiksi. DICOM-olio voidaan ajatella puumaisena hierarkiarakenteena. Ylimpänä hierarkiassa on DICOM IOD-olio, josta haarautuu erilaisia ominaisuuksia. IOD-olion hierarkiaa alaspäin mentäessä vastaan tulee erilaisia alemman tason ominaisuuksia, jotka voivat olla joko erilaisia IE-entiteettejä tai moduuleita. Lopulta alaspäin mennessä päädytään varsinaiseen tutkimuksen kannalta tärkeään binaari- tai merkkimuotoiseen sisältödataan, esimerkiksi kuviin tai potilaan henkilötietoihin.

IOD-hierarkian ylimmällä tasolla ovat niin sanotut Normalized ja Composite IOD -oliot. Normalized-oliot ovat tosielämää vastaavia kokonaisuuksia ja Composite-oliot yhdistelmä tosielämää vastaamattomista ja vastaavista ominaisuuksista. Esimerkki Normalized-oliosta on Patient IOD, joka sisältää tietoja potilaasta, kuten esimerkiksi henkilötunnuksen, nimen ja niin edelleen. Esimerkki Composite-oliosta on CT image IOD, joka sisältää sekä tosielämään kuuluvaa dataa (esimerkiksi varsinainen kuva), sekä tosielämään kuulumattomia ominaisuuksia (esimerkiksi kuvan leveys kuvapisteinä).

3.2 Information Object Definition

Erityyppisillä IOD-olioilla on eri ominaisuuksia. DICOM-standardi määrittelee kaikki mahdolliset IOD-tyypit ja niiden ominaisuudet erillisissä taulukoissa. Esimerkki kyseisestä ominaisuusluettelosta on taulukossa 1. IOD koostuu attribuuteista, jotka ovat DICOMissa nimetty nimellä Information Entities (IE). Information Entityt taas koostuvat Information Moduleista. Alla olevassa kuvassa 1 kuvataan UML-luokkakaavion avulla IOD-dataolion rakenne.



Kuva 1. UML-luokkakaavio IOD-oliosta ja moduuleista.

Erilaisten potilastutkimusten IOD-olioiden kuvaukset löytyvät DICOM-standardin osion PS 3.3 [2011] tauluista. Kyseisen kappaleen tauluissa kuvataan kaikki pakolliset ja valinnaiset attribuutit. Tosielämää vastaavien olioiden taulujen rakenne on erilainen verrattuna tosielämää vastaamattomiin. Seuraavassa taulukossa 1 on esimerkki Normalized IOD -oliosta eli tosielämää vastaavasta oliosta. Taulun kuvaus sisältää määrittelytaulukon A.8-1 eli IOD-olion Secondary Capture (SC) Image -kuvalle.

IE	Module	Reference	Usage
Patient	Patient	C.7.1.1	M
	Clinical Trial Subject	C.7.1.3	U
Study	General Study	C.7.2.1	M
	Patient Study	C.7.2.2	U
	Clinical Trial Study	C.7.2.3	U
Series	General Series	C.7.3.1	M
	Clinical Trial Series	C.7.3.2	U
Equipment	General Equipment	C.7.5.1	U
	SC Equipment	C.8.6.1	M
Image	General Image	C.7.6.1	M
	Image Pixel	C.7.6.3	M
	Device	C.7.6.12	U
	Specimen	C.7.6.22	U
	SC Image	C.8.6.2	M
	Overlay Plane	C.9.2	U
	Modality LUT	C.11.1	U
	VOI LUT	C.11.2	U
	ICC Profile	C.11.15	U
	SOP Common	C.12.1	M

Taulukko 1. Kuvaus A.8-1 Secondary Capture (SC) Image -kuvan IOD-oliosta

Taulukon 1 otsikkojen kuvaus on seuraavassa:

- IE - Information Entity on IOD-olion attribuutti ja se sisältää erilaisia moduuleita.
- Module - Jokainen IE sisältää yhden tai useampia moduuleita, jotka taas sisältävät varsinaista tutkimusdataa.
- Reference - Tarkoittaa viittausta DICOM-määrittelyn PS 3.3 [2011] osioon C, joka sisältää kuvauksen kyseisestä moduulista sekä siihen liittyvät tagit.
- Usage - Kentän pakollisuutta kuvaava arvo. M (Mandatory) tarkoittaa pakollista tietoa. U (User Option) tarkoittaa valinnaista tietoa. C (Conditional) tarkoittaa ehdollisesti pakollista arvoa.

Esimerkiksi taulukosta 1 voidaan tulkita, että SC IMAGE sisältää pakollisena tietona potilaan tunnistetiedot, koska Patient-module on pakollinen (M). Kyseinen kuvaus sisältää muitakin pakollisia tietoja (M), kuten tutkimuksen tietoja (General Study Module) sekä kuvan kuvapistedatan (Image Pixel Module).

Composite IOD määrittely poikkeaa taulukosta 1 niin, että siinä ei ole IE tai Usage-kenttiä. IE-entiteetin puuttuminen johtuu siitä, että Composite IOD -oliot sisältävät vain yhden reaali maailman olion attribuutteja. Pakollisuutta ei pystytä myöskään määrittelemään Composite-olioissa, koska pakollisuus riippuu käytettävästä palvelusta.

IOD-oliot voidaan ajatella seuraavassa kappaleessa esiteltävien dataelementtien kokoelmana. Moduulit ovat standardin määrittelemiä yhteen koottuja PS 3.3 [2011]-osion määrittämiä kokonaisuuksia. Moduulit sisältävät samaan ominaisuuteen liittyviä dataelementtejä, kuten taulukosta 1 nähdään. Tästä esimerkkejä ovat potilaan perustiedot, henkilötunnus, potilaan nimi, äidin nimi ja niin edelleen, jotka kootaan samaan moduuliin.

Seuraavissa osioissa kuvataan, kuinka IOD-olion sisältämä data luodaan VR-koodauksen ja sanakirjan avulla.

3.3 VR-tietotyypit

VR tarkoittaa datan tietotyyppiä. Erilaisia tietotyyppiä voivat olla esimerkiksi merkkijono, aika, numerot tai erilaiset binaaridatat. Edellisessä osiossa kuvatut IOD- oliot sisältöineen koodataan aina VR-tietotyyppien avulla. VR-tietotyyppiä on 27 kappaletta ja jokaisella tietotyyppillä on seuraavat ominaisuudet:

- Lyhenne ja nimi - VR-tietotyypin nimi ja nimen lyhenne, jotka koostuvat kahdesta kirjaimesta.
- Kuvaus sisällöstä - Kuvaus siitä, millaista dataa kyseinen tietotyyppi sisältää.
- Sallitus merkit - Kuvaus siitä, millaisia merkkejä tietotyyppi voi sisältää.
- Pituus - Kuinka pitkä datakenttä on. Pituus ilmoitetaan joko tarkkana pituutena tai maksimipituutena.

Standardin VR-taulu osiossa PS 3.5 [2011] kokoaa kaikki mahdolliset tietotyypit, joiden avulla data voidaan koodata. Seuraavassa taulukossa 2 on esimerkkejä kyseisestä taulukosta ja erilaisista VR-tietotyypeistä.

Lyhenne ja nimi	Kuvaus sisällöstä	Sallitut merkit	Pituus
LO Long String	Merkkijono.		Enintään 64 merkkiä
DA Date	Merkkijono muodossa YYYYMMDD, esimerkiksi 20090325	0-9	8
SL Signed Long	Binaarimuotoinen komplementtiluku		4

Taulukko 2. Esimerkkejä VR-tietotyypeistä

Erilaisia merkkijonotyyppisiä VR-tietotyyppisiä ovat CS (code string), SH (short string), LO (long string), ST (short text), LT (long text) ja UT (unlimited text). Nämä ovat yksinkertaisia tietotyyppisiä ja niiden käsittely on suoraviivaista. Päivä ja aikaa kuvaavat tietotyypit ovat DA (päivämäärä), TM (aika), DT (aika ja päivämäärä), AS (ikä). Ajan muoto ilmoitetaan muodossa HHMMSS ja päivämäärä muodossa YYYYMMDD. Yksi erikoinen tietotyyppi on UN (tuntematon), joka on tarkoitettu arvolle, jonka sisältöä ei voida muilla VR-tietotyypeillä määrittellä.

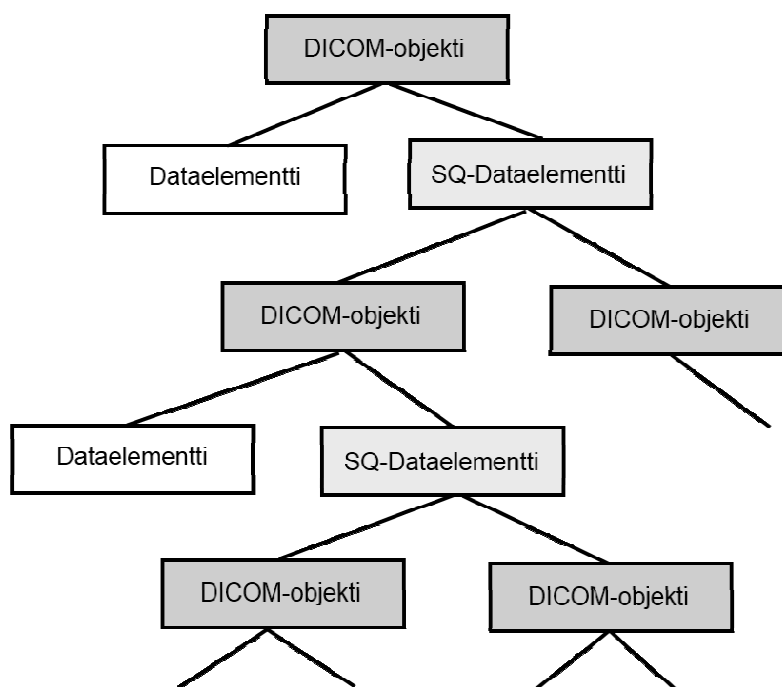
Jokaisen datakentän täytyy olla pituudeltaan parillinen. Tämä tarkoittaa, että varsinaista arvoa on tarvittaessa manipuloitava, jotta tämä sääntö toteutuisi. Tämä tehdään sekä binaari-, että tekstimuotoisille kentille. Jos merkkimäärä on todellisuudessa pariton, niin manipulointi tehdään niin, että tiedon perään lisätään yksi tavu. Jos kenttä on merkkijono, sen sisältämien merkkien määrän on oltava parillinen. Jos taas kyseessä on binaarimuotoinen kenttä, kentän tavumäärän pituus on oltava parillinen. Merkkijonotyyppisen datakentän perään lisätään tarvittaessa yksi tyhjä (space) merkki ja binaarimuotoisen datakentän arvoon yksi NULL-tavu. Parillisen pituuden sääntö pätee aina, eli myös silloin, kun kentän datan pituus on muuttuva.

DICOM-olion VR-data voi olla koodattu joko implisiittisesti tai eksplisiittisesti. Eksplisiittinen VR-koodaus tarkoittaa sitä, että jokaisen elementin kuvauksessa itsessään on mukana käytetyn VR-tietotyypin lyhenne, esimerkiksi "LO". Implisiittisessä koodauksessa tätä lyhennettä ei ole datassa itsessään. Tällöin luotetaan siihen, että vastaanottava pää osaa tulkita dataa PS 3.5-osion taulukon perusteella. Eksplisiittisessä koodauksessa dataolion elementit ovat muutaman tavun pidempiä kuin

implisiittisessä. Yksittäisessä DICOM-olioissa on aina käytettävä vain yhtä koodaustapaa, molempia ei voida käyttää saman olion sisällä. Useimmiten on turvallisinta käyttää eksplisiittistä koodausta [Pianykh 2009].

3.4 SQ VR, sisäkkäiset dataoliot

SQ (Sequencing Data Sets) -tyyppinen elementti on muihin elementteihin verrattuna erilainen, koska se voi sisältää useita sisäkkäisiä VR-elementtejä. Tämän avulla voidaan luoda tarvittaessa monimutkaisia rakenteita. SQ-elementtiin voi upottaa kokonaisen DICOM-olion. Tällöin yksittäinen olio saattaa sisältää useita olioita. Dataolio voi aina sisältää tällaisen SQ-elementeillä kootun puurakenteen, jonka haarat voivat taas sisältää lisää elementtejä tai sitten varsinaista sisältödataa. Useita DICOM-olioita sisältävää SQ-rakennetta havainnollistetaan kuvassa 2.



Kuva 2. Syväälle haarautuva SQ-elementtirakenne

SQ on VR-tietotyyppinen elementti. Muidenkin VR-elementtien tapaan sen koodaukseen käytetään seuraavassa osiossa kuvattua sanakirjaa. SQ VR -elementit ovat varsin käyttökelpoisia esimerkiksi videodatassa. DICOM BMP -standardin mukainen videodata koostuu useista peräkkäisistä kuvista, jotka vastaanottavassa päässä koostetaan yhdeksi esitykseksi datassa määrättyllä kuvanopeudella. Kuvadatasta kerrotaan myöhemmin tässä kappaleessa.

3.5 Sanakirja ja tagiryhmät

Yksittäisten DICOM-elementtien tulkitsemiseen tarvitaan aina NEMA PS 3.6 [2011]-osion tarjoamaa sanakirjaa. Erilaisia tageja on tuhansia ja ne kaikki määritellään kyseisessä sanakirjassa. Alla olevassa taulukossa 3 on esimerkkejä potilaan tietoihin liittyvistä sanakirjamerkinnoistä.

Tagi	Nimi	Avainsana	VR	Kerroin
(0010,0010)	Patient's Name	PatientName	PN	1
(0010,0020)	Patient ID	PatientID	LO	1
(0010,0040)	Patient's Sex	PatientSex	CS	1

Taulukko 3. Esimerkkejä potilaan sanakirjamerkinnoista.

Taulukon 3 otsikkojen kuvaus on seuraavassa:

- Tagi - Datan tyyppiä kuvaava binaarikoodi. Tagi ilmoitetaan aina jokaisen dataelementin alussa. Dataelementin sisältö tunnustetaan tämän binaarikoodin avulla.
- Nimi - Tagin nimi.
- Avainsana - Yksilöivä avainsana tagille.
- VR - Datan tietotyyppi, jota kuvattiin aikaisemmin.
- Kerroin - Lukumäärä, montako kertaa tämä tagi voi olla yhdessä DICOM-dokumentissa.

Kaikki tagit kuuluvat johonkin samantyyppisistä tiedoista koostuvaan ryhmään. Tagiryhmää kuvaa tagin ensimmäinen binaaritavu, kuten esimerkiksi taulukossa 3 ryhmä on 0010. Jotkut tagiryhmät ovat varattuja. Esimerkiksi (0008)-alkuiset tagit ovat varattuja tietoliikenteen käyttöön ja taulukossa 3 esiintyvät (0010)-alkuiset tagit potilaan tietoihin. DICOM-olioissa tagit tulevat aina järjestyksessä pienimmästä suurimpaan.

Varsinaisessa DICOM-datassa saman tietoryhmän tagit tulevat aina peräkkäin. Tämä ominaisuus kokoaa eri ryhmät aina yksittäisiksi kokonaisuuksiksi. Ryhmät tulevat myös aina nousevassa järjestyksessä, esimerkiksi 0008-ryhmän tagit ovat aina ennen 0010-ryhmän tageja. Myös jokaisen ryhmän yksittäiset tagit tulevat nousevassa järjestyksessä. Esimerkiksi tagi (0010,0020) tulee ennen tagia (0010,0040).

Ennen jokaista tietoryhmää ilmoitetaan 0000-alkuisella tagilla ryhmän sisältämien elementtien yhteinen tavumäärä. Esimerkiksi potilaan ryhmän koko ilmoitetaan tagissa (0000,0010). Tämä on hyödyllinen tieto, koska sen avulla voi esimerkiksi jättää kyseinen tavumäärä lukematta, jos lukijalla ei ole tarvetta kyseiselle tiedolle.

3.6 Dataelementit

Varsinaiset dataelementit perustuvat aikaisemmissa osioissa kuvattuihin DICOM-sanakirjan määrittelemiin tageihin ja VR-koodaukseen. Yksittäinen dataelementti sisältää varsinaista tutkimusdataa, kuvia, potilaan tietoja tai muuta tutkimukseen liittyvää tietoa. Yksittäisellä dataelementillä on seuraavat ominaisuudet:

- **Tagi** - Tagikirjasto sisältää binaarimuotoisen tagin jokaiselle mahdolliselle tiedolle, jota DICOM-standardin avulla voidaan välittää. Esimerkiksi aikaisemmassa osiossa mainittu (0010,0010) sisältää potilaan nimen.
- **VR** - datan tietotyyppi tekstimuodossa. Tämä on mukana vain eksplisiittisesti koodatussa VR-oliossa (tästä kerrottiin osiossa 3.3).
- **Pituus** - Kuinka monta merkkiä varsinainen datasisältö sisältää. Pituuden tulee aina olla parillinen luku ja se ilmoitetaan binaarilukuna.
- **Arvo** - Varsinainen datasisältö. Esimerkiksi potilaan nimi voisi olla Matti^Meikäläinen(20). Tyhjä merkki (20) lisätään tässä arvon loppuun, jotta siitä saadaan pituudeltaan parillinen (tästä kerrottiin osiossa 3.3).

Nämä ominaisuudet ilmoitetaan jokaisen dataelementin kohdalla. Seuraavaksi esitellään, kuinka tämä tapahtuu käytännössä. Alla olevassa esimerkissä kuvataan, kuinka potilaan nimi koodataan DICOM-dataelementtinä. Esimerkin binaaritavut on muunnettu merkkimuotoisiksi datan muodon tulkitsemisen helpottamiseksi. Yksittäinen tavu ilmoitetaan sulkeissa, esimerkiksi (20). Alla olevan esimerkkielementin binaaridata esitetään Little Endian -muodossa ja kyseessä on eksplisiittisesti koodattu dataelementti. DICOM-standardissa kaikkien arvokenttien on oltava pituudeltaan parillisia ja koska alla olevan esimerkin nimi on pituudeltaan 17 merkkiä välimerkkeineen, loppuun lisätään yksi tyhjä tavu. Tavujärjestyksestä puhutaan lisää osiossa 3.9 ja parillisuussääntöjä käsitellään myöhemmin kappaleessa 5.5.

(10) (00) (10) (00) PN (12) (00) Meikäläinen^Matti(20)

Kyseisessä elementissä ilmoitetaan ensin sanakirjan tagi (10) (00) (10) (00). PN tarkoittaa VR-tietotyyppiä. VR ilmoitetaan, koska kyseessä on eksplisiittinen koodaus. VR myös kuvaa, millaista dataa (binaari- vai merkkimuotoista) seuraavaksi on tulossa. Jos kyseessä olisi eksplisiittinen koodaus, VR-tietotyyppi ja datan merkki/binaarimuoto pääteltäisiin tagista. VR-tietotyypin jälkeen ilmoitetaan elementin arvon pituus, joka on 18 merkkiä eli binaarilukuna (12) (00). Nimen ilmoittaminen ja ^-merkki kuuluvat standardin tapaan erotella nimet toisistaan. Tämä on määritelty standardin toimesta tarkasti kaikkien etunimien osalta. Näitä ohjeita ei kuitenkaan useinkaan noudateta [Pianykh, 2008]. Useimmiten voidaan kuitenkin olettaa, että sukunimi tulee ensin, sitten

^-merkki ja sitten ensimmäinen etunimi. Esimerkin lopussa on binaariluku (20) tekemässä elementin arvon pituudesta parillisen.

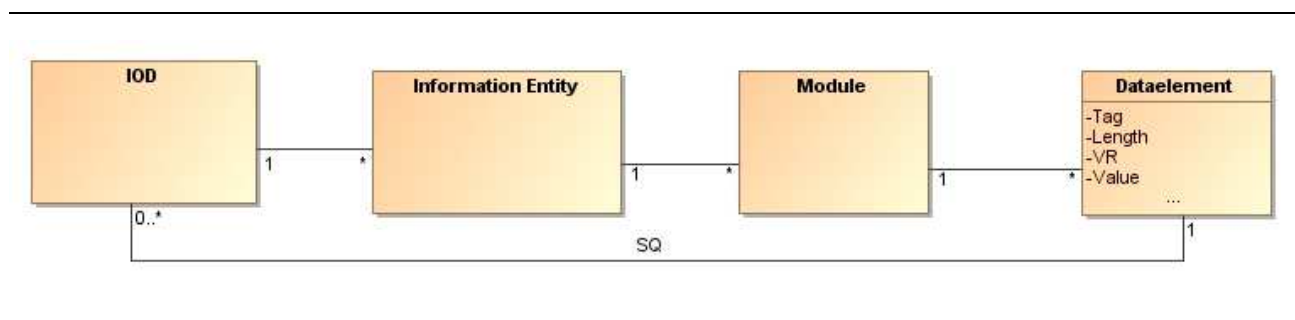
3.7 Unique Identifiers (UID)

DICOM-olioissa sisältödatan muuttaminen luo aina uuden instanssin oliosta. Lisäksi jos tutkimuksen kuvasta otetaan kopio, niin instanssi vaihtuu vaikka kuvadata ei muuttuisi. Uutta kuvaa voidaan rajata, sen värejä voidaan muuttaa tai siihen voidaan lisätä huomautuksia. Tällöin kuvasta tulee erilainen kuin alkuperäisestä ja silti se on kopio alkuperäisestä. Radiologin tulee pystyä erottamaan eri instanssit toisistaan kaikissa edellä mainituissa tilanteissa. Tässä apuna on UID (Unique Identifiers), jonka avulla voidaan tunnistaa, mikä versio kuvasta on kyseessä.

UID on yleensä pisteillä erotettu numerosarja ja se voi näyttää esimerkiksi tältä: 1.2.543.10098.2.4. Jokaisen UID-arvon on tarkoitus olla globaalisti uniikki, jotta kuvaa ei sekoitettaisi esimerkiksi toisen sairaalan kuvaan. Tämän vuoksi UID-tunnisteeseen liitetään alkuun organisaation juuritunnus seuraavasti: <organisaation juuri>.<loppuliite>. Standardi määrittelee, että jokaisen DICOM-standardia käyttävän organisaation tulisi hakea NEMA-järjestöltä oma yksilöivä juuri-ID [NEMA, PS 3.5, 2011].

3.8 Dataelementit IOD-olioissa

DICOM-oliot koostuvat useista peräkkäisistä VR-dataelementeistä. Tämä sääntö pätee kaikkiin muihin DICOM-olioihin paitsi osiossa 3.4 kuvattuun SQ VR -tyyppiseen elementtiin, joka sisältää kokonaisia dataolioita. Alla olevassa kuvassa 3 kuvataan, kuinka aikaisemmin kuvatut standardin PS 3.3 [2011] Information Object Definitions -osion Normalized tai Composite IOD -dataelementit sisältävät näitä dataelementtejä. IOD-määrittely ja moduulien määrittely yhdessä kertovat, että mitkä dataelementit ovat pakollisia kyseisissä olioissa. Kuvassa 3 havainnollistetaan myös sitä, kuinka dataelementti voi sisältää useita olioita SQ VR -tietotyyppin avulla.



Kuva 3. VR-dataelementin suhde dataolioon.

3.9 Binaaridata

DICOM VR -datan binaarimuotoiset tagit tuottavat tiettyjä haasteita, joita merkkijonomuotoisessa elementtirakenteessa (kuten XML) ei tarvitse huomioida. Binaaridatan käsittelyssä täytyy aina ottaa huomioon tavujärjestys. Binaaritavuilla on aina kaksi mahdollista tavujärjestystä, Big Endian ja

Little Endian. Big Endian -tavujärjestyksessä ensimmäiseksi ilmoitetaan aina merkitsevä tavu, esimerkiksi 0x7F00. Little Endian -tavujärjestyksessä taas merkitsevä tavu ilmoitetaan vasta jälkimmäisenä: 0x007F. Tavujärjestys saattaa riippua esimerkiksi käyttöjärjestelmästä. Big Endian -tavujärjestystä käyttää muun muassa Mac-käyttöjärjestelmä ja Little Endian -tavujärjestystä käyttää Microsoft Windows.

Dataelementtien tagien luvun tekee haasteelliseksi se, että oliot saattavat olla keskenään eri tavujärjestyksellä koodattuja jopa yksittäisten olioiden sisällä. Tästä on esimerkkinä sisäkkäiset SQ-elementit, jotka voivat sisältää eri tavujärjestyksen sisältäviä olioita. Tämä on kuitenkin harvinaisempi tilanne. Varsinaisten binaarityyppisten VR-dataelementtien tavujärjestys ilmoitetaan jokaisen elementin kuvauksen ohessa. Binaarikoodattua dataa ovat esimerkiksi kuvat, josta kerrotaan seuraavaksi.

3.10 Kuvadata

Vaikka DICOM-standardilla voidaan siirtää Structured Reporting -palvelun (kuvataan kappaleessa 4.11) avulla millaista tutkimukseen liittyvää dataa tahansa, suurin osa DICOM-verkossa kulkevasta liikenteestä on kuvadataa. Standardi on alun perin kehitetty kuvien arkistointiin ja esimerkiksi tarpeeseen välittää kuvadataa eri ohjelmistojen välillä.

DICOM määrittelee oman DICOM BMP -kuvamuodon, joka on kuvien oletussiirtomuoto. DICOM BMP kuvadata on VR-koodattua muiden tietojen tapaan. DICOM BMP -muodon varsinainen kuvadata ja sen kannalta oleelliset tiedot löytyvät DICOM-sanakirjan määräämistä tageista. DICOM BMP -datassa varsinainen kuvadata siirretään (7FE0, 0010)-tagin sisällä. Kuvadatan VR-tietotyyppi on binaarimuotoinen ja se on aina joko OW tai OB. OB-tyyppisessä datassa yksi kuvapiste ilmoitetaan yhdellä tavulla ja OW-datassa kahdella tavulla. OW-tyyppistä kuvadataa käytetään yleisesti suurimmassa osassa sairaalan kuvantamislaitteista, esimerkiksi röntgen- ja magneettikuvissa [Pianykh 2009]. Kuvadataan kuuluu (7FE0, 0010)-tagin lisäksi muita kuvadataa selittäviä tageja, kuten esimerkiksi näytettä per kuvapiste. Yhden kuvapisteen käyttämälle bittimäärälle on myös oma taginsa.

Sairaalajärjestelmien välillä liikuteltavat kuvat ovat yleensä harmaasävyisiä, jolloin kuvadata sisältää yhden näytteen per kuvapiste. Harmaasävykuvan VR-tyypin ollessa OW eri harmaasävyjä voi olla $2^{16} = 65\,536$ kappaletta. Jos kuvapisteeseen käytetään 8 bittiä, harmaasävyjä voi yhdessä kuvapisteessä olla $2^8 = 256$ kappaletta. Sen sijaan jos kuvapisteessä on 10 bittiä, yhdessä kuvapisteessä voi olla harmaasävyjä $2^{10} = 1024$ kappaletta. Kuvapisteen viemä tila varataan kuitenkin todellisuudessa niin, että yksi kuvapiste varaa aina yhden tavun verran bittejä. Esimerkiksi 10 bittiä vie todellisuudessa 16 bittiä. Todellinen kuvapisteen viemä bittimäärä ilmoitetaan vielä erikseen omassa tagissaan.

DICOM BMP -muodon värisävyjen tasatavun säännön takia kuvasta saattaa jäädä ylitse muutamia bittejä jokaista kuvapistettä kohti. DICOM-standardissa on myös keino hyödyntää näitä tarpeettomaksi jääneitä bittejä. Niihin voidaan myös lisätä tarvittaessa tutkimusdataa. Vaikka ylitse jäävien bittien määrä saattaakin suurissa kuvissa nousta merkitseväksi, niistä saatava hyöty on

kuitenkin yleensä varsin pieni ja kuvanpakkauksella saavutettava etu on usein ylivoimainen verrattuna tähän ylitsejäävän datan hyötykäyttöön.

Kuvien välityksen ohella DICOM-datassa on usein mukana videodataa. Esimerkiksi ultraäänitutkimus sisältää useimmiten videodataa. DICOM-standardin video-ominaisuus on toteutettu niin, että videodata toimitetaan DICOM-olion sisällä sarjana peräkkäisiä kuvia. Olio on tallennetut kuvat esitetään katseluohjelmassa peräjälkeen erikseen määritetyllä kuvanopeudella (Frame Rate). DICOM-standardi määrittelee erilliset tagit sekä kuvanopeudelle, että muille videodatan ominaisuuksille.

3.11 Kuvanpakkaus

DICOM BMP on ainoa kuvamuoto, jonka tuen standardi määrää pakolliseksi. DICOM BMP -muodon lisäksi DICOM kuitenkin tukee joitakin suosituimpia kuvanpakkausalgoritmeja. DICOM-datayhteyden muodostamisessa neuvotellaan se, mitä pakkausmuotoa välitettävässä datayhteydessä käytetään. Jos vastapää ei tue pakkausmuotoa jota lähettäjä haluaisi toimittaa, on lähettäjän varauduttava siihen, että kuva on välitettävä DICOM BMP -muodossa. Tämä johtaa toisinaan siihen, että kuvien siirto DICOM-verkossa estyy koska lähettäjä ei pysty muuntamaan kuvaa neuvottelussa takaisin alkuperäiseen DICOM BMP -muotoon. DICOM BMP on monista syistä sekä oletus, että yleisin pakkaustyyppi [Pianykh, 2008]. Datayhteyden muodostaminen ja kuvapakkauksista neuvottelu kuvataan tarkemmin seuraavassa kappaleessa.

DICOM BMP -muodossa on kuvadatan vaatima tavumäärä yleensä huomattavasti suurempi kuin kuvanpakkausalgoritmilla käsitellyssä kuvassa. Esimerkiksi yksi CT-kuvasarja saattaa viedä pakkaamattomana 100 megatavua [Pianykh, 2008]. Levytilan kannalta tämä ei ole välttämättä suuri ongelma, mutta verkon kapasiteetin kannalta kuvadatan koko on kriittistä. Usein lääkärin täytyy potilaan diagnoosia tehdessään odotella minuutteja sitä, kuinka isokokoista kuvadataa ladataan arkistosta. Ratkaisuksi tähän on esitetty niin sanottua Streaming-ominaisuutta, joka on määritelty JPEG 2000 -kuvanpakkausstandardissa oletuksena. Streaming tarkoittaa, että kuvasta ladataan aina vain osia, joihin käyttäjä on suurentanut eli niin sanottuja Regions of Interest (ROI) -kohtia. Streamingissa ladataan aluksi näytön kuvapisteen määrään sopiva kuva. Kun kuvaa suurentaa, niin ladataan aina uusi kuva, joka sisältää mahdollisimman tarkan kuvan niin, että kuvapisteen määrä vastaa aina vastapään näytön kuvapisteen määrää. Streaming on varsin kätevä ominaisuus, koska joissain tilanteissa se nopeuttaa kuvien selaamista huomattavasti. Streamingiä kuitenkin pidetään toisinaan käyttäjien mielestä hieman kankeana, koska jokainen suurennus aiheuttaa aina pienen viiveen. Sairaanhoidon prosessiin näyttää sopivan paremmin se, että koko tutkimusdata ladataan aluksi yhdellä kertaa. Tällöin pieniä viiveitä diagnoosia tehdessä ei ole, koska kuvadata on ladattu kokonaan [Pianykh 2009].

Häviöttömän JPEG 2000 -kuvanpakkausalgoritmin käyttö on yleistynyt nykyään DICOM-verkoissa. Pelkkää JPEG 2000 -kuvanpakkausta käyttäviä verkkojakin on suunniteltu. Näitä verkkoja kutsutaan toisinaan nimellä DICOM2000 [Ivetic & Dragan, 2009].

Kuvien arkistoinnissa on tärkeää huomioida se, onko kuvanpakkausalgoritmi häviötön vai häviöllinen. On tärkeää, että kuvadata säilyy alkuperäisenä. Häviöllisessä pakkauksessa kuvan laatu huononee joka kerta kun sitä pakataan. Tällöin kuvan jatkuva muokkaus ja tallennus muuttaa kuvan helposti rakeiseksi. Häviöllisten tiedonpakkausalgoritmien käyttö arkistoinnissa ei olekaan useimmiten suotavaa.

Kuvanpakkaukseen liittyy muitakin ongelmia. Monet kuvanpakkausalgoritmit eivät tue DICOM BMP -muodon harmaasävyjen määrää. Esimerkiksi häviöllisessä JPEG-pakkauksessa harmaasävyjen määrän vähentäminen huonontaa kuvan laatua oleellisesti. DICOM ei myöskään välitä tietoa kuvanpakkauksen asetuksista. Kuvan vastaanottaja ei esimerkiksi voi tietää, onko kuva pakattu häviöllisellä vai häviöttömällä asetuksella. Myös tämä saa ohjelmistojen valmistajat usein suosimaan DICOM BMP -muotoa.

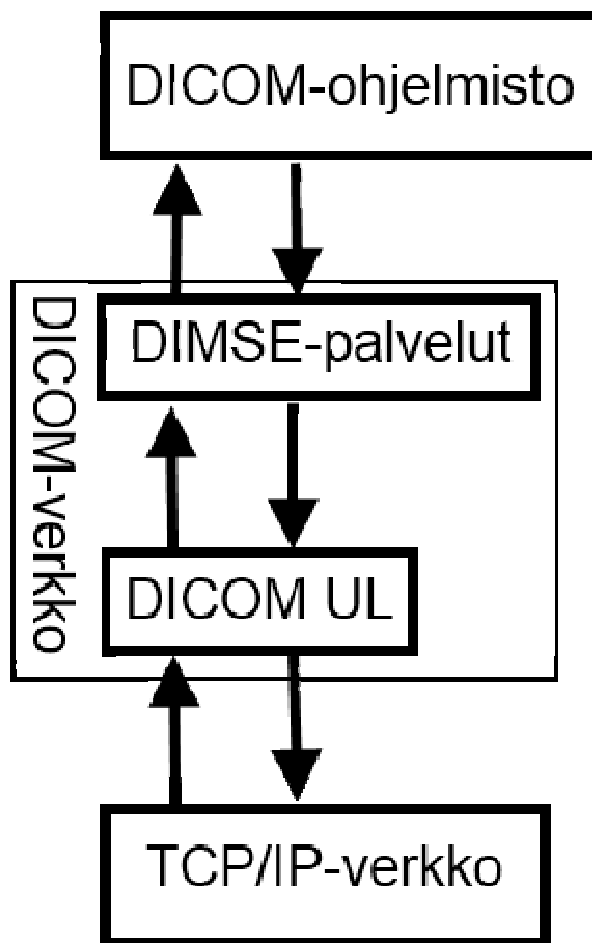
Tämä kappale kuvasi DICOM-muotoisen datan olioista yksittäisten VR-dataelementtitasolle ja kuvadataan. Seuraavaksi kuvataan, kuinka DICOM-muotoisia olioita siirretään tietoverkossa.

4. DICOM-tietoliikenne

Tässä kappaleessa kuvataan ensin protokollan ylemmän tason DIMSE (DICOM Message Service Elements) -verkkoliikennepalvelut sekä palvelut ja datan yhdistävä SOP-luokka. Lisäksi kuvataan erilaiset roolit, jotka verkko asettaa keskusteleville osapuolille. Palveluluokkien kuvauksen jälkeen kuvataan TCP/IP-protokollan päällä toimiva alemman tason verkkoliikenne Associations, eli toiselta nimeltään DICOM UL (Upper Layer Protocol). DICOM UL suorittaa varsinaisen keskustelun ohjelmistojen välillä. Osiossa kuvataan, kuinka ohjelmistot ottavat DICOM-verkossa yhteyden, kuinka dataa siirretään ja kuinka yhteys katkaistaan. Protokollan toimintaa kuvataan esimerkiksi sekvenssikaavioiden ja FSP-prosessikaavion avulla. DICOM UL kuvauksen jälkeen esitetään eri SOP-luokkien toiminnot muun muassa FSP-prosessikaavioiden avulla. Tämä kappale on koottu pääasiassa DICOM-standardin dokumentaatiosta. Sen sijaan mukava olevat FSP-prosessikaaviot ja LTS-diagrammit on luotu liitteen 1 FSP-kuvauksen avulla.

4.1 DICOM-verkkoliikenteen hierarkia

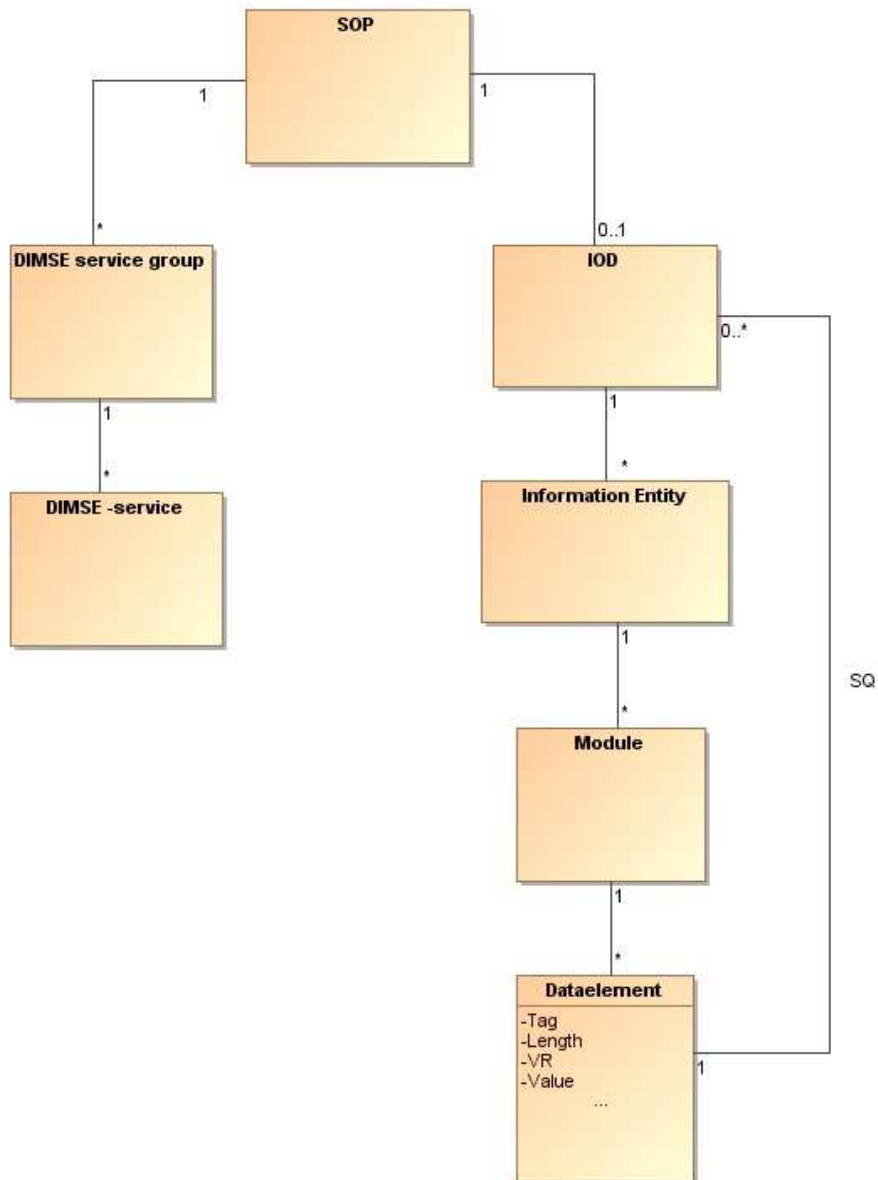
DICOM-standardi sisältää kaksitasoisen tietoliikennehierarkian. Ylemmällä tasolla sijaitsevat erilaiset verkkoliikennepalvelut ja alemmalla tasolla varsinainen keskustelu TCP/IP-verkon välityksellä. Kuvassa 4 havainnollistetaan DICOM-standardin toimintatapaa verkkoliikenteen ja ohjelmiston näkökulmasta. Kuvassa Upper Layer Protocol on selkeästi erotettu tässä kappaleessa käsiteltävistä ylemmän tason palveluista. Upper Layer Protocol on pelkästään datan siirtoon paikasta A paikkaan B keskittyvä protokolla. Sen sijaan SOP-luokka on yhdistelmä DICOM-dataolioista ja erilaisista palveluista, jotka liittyvät kyseisiin dataoloihin.



Kuva 4. TCP/IP-verkon päällä toimivan DICOM-verkko.

4.2 SOP-luokka

SOP (Service-Object Pair) on yhdistelmä DIMSE-palvelusta ja aikaisemmin kuvatusta IOD-dataoliosta. DIMSE-palvelut ovat pyyntöjä SOP-luokan avulla yhdistetyn IOD-dataolion käsittelemiseen. Toisin sanoen DIMSE pyytää tekemään IOD-dataoliolle jonkun toiminnon. Esimerkiksi C-MOVE -kutsussa ilmoitetaan IOD-oliona pelkkä tutkimuksen UID, joka halutaan siirtää ilmoitettuun määränpäähän. Joissain SOP-luokissa ei ole mukana IOD-dataoliota. Esimerkiksi C-ECHO -pyyntö ei sisällä varsinaista dataoliota, vaan pelkän vastauksen saaminen tulkitaan aina positiiviseksi vastaukseksi. Alla olevassa kuvassa 5 kuvataan SOP-luokan suhde DIMSE-palveluryhmiin ja IOD-dataoloihin UML-luokkakaavion avulla.



Kuva 5. UML-luokkakaavio, johon on lisätty SOP-luokka ja DIMSE-palvelut.

Standardi kuvaa Conformance-osiossa [NEMA, PS 3.2, 2011], kuinka luoda Conformance Statement -dokumentti ja kuinka sitä tulee noudattaa vastapuolen toimesta. Conformance Statement perustuu paljolti ohjelmiston toteuttamien SOP-luokkien kuvaamiseen, joten SOP-luokkien merkitys siltä osalta on tärkeä. Conformance Statement -dokumentissa kuvataan ohjelmiston käyttämät Service Class -luokat, Information Object -dataoliot ja protokollat, jotka kyseinen ohjelmisto on toteuttanut. DICOM-ohjelmiston täytyy toimittaa kyseinen dokumentti vastapuolelle, jotta se voi saada hyväksynnän DICOM-yhteensopivuudesta.

Seuraavaksi tässä kappaleessa lähdetään kuvaamaan erilaisia palveluita. Ensin kuvataan yksilöinti ja käyttöroolit, joita tarvitaan palveluiden käytössä. Sitten kuvataan DIMSE-palvelut ja niiden tarkoitus ja sen jälkeen DICOM UL eli protokolla, jolla ohjelmistot keskustelevat keskenään.

4.3 Yksilöinti DICOM-verkossa

Jokainen ohjelmisto on yksilöitävä DICOM-verkossa. Yksittäinen ohjelmisto yksilöidään DICOM-verkossa seuraavien tietojen perusteella:

- AE Title
- IP-osoite
- Portti

IP-osoite ja portti tarkoittavat TCP/IP-protokollan määrittelemää yhteyskanavaa. AE tarkoittaa DICOM-verkkoon kytkettyä ohjelmaa, jonka kanssa halutaan keskustella. AE Title tarkoittaa kyseisen ohjelman yksilöivää nimeä ja se on maksimissaan 16-merkkinen. Jokaisella DICOM-verkon AE-entiteetillä on oltava oma pysyvä IP-osoite.

DICOM-verkossa on aina enemmän kuin yksi AE. AE-entiteettejä voivat olla esimerkiksi DICOM-palvelin, tulostin, modaliteetti (kuvantamislaite) tai vaikka DICOM-verkkoon yhdistetty työasemaohjelmisto. AE tarkoittaa siis yksittäistä ohjelmaa, eikä esimerkiksi koko työasemaa. Aikaisemmin DICOM-standardi määritteli NEMA PS 3.9 [2011]-osiossa oman tietoliikenneprotokollan, mutta nykyään verkkona käytetään Internetin kautta yleistynyttä TCP/IP-protokollaa. Koska standardi toimii TCP/IP-verkon päällä, jokaisella AE-entiteetillä on oltava oma verkkokortti ja sitä kautta IP-osoite. AE Titlet yksilöivät ohjelmistot DICOM-verkossa, mutta verkkoliikenteen kannalta porttien avulla työasemalle voidaan asentaa useita eri ohjelmistoja. Työasemassa voi olla esimerkiksi tulostin ja työaseman ohjelmisto voivat olla oma AE-entiteettinsä. Tällöin jokaiselle AE-entiteetille asetetaan eri portti.

DICOM-verkossa ohjelmistojen väliseen keskusteluun käytetään porttia, joka on väliltä 0-65 535. DICOM-standardi määrittelee oletusportiksi 104, mutta porttiasetukset voidaan määrittellä ohjelmistokohtaisesti. Kaikkien verkon keskusteluosapuolien on kuitenkin käytettävä samaa porttia. On myös tärkeää, ettei käytetä sellaista porttia, joka on varattu toiselle ohjelmistolle. Esimerkiksi portti 80 on yleensä selainohjelmistojen käytössä joten sen käyttö ei ole suotavaa.

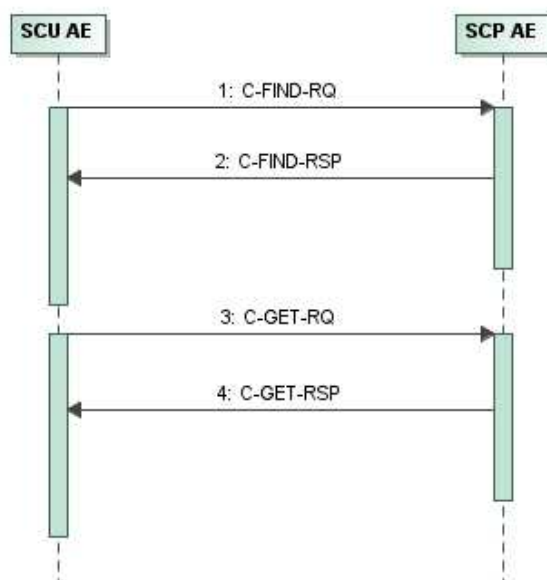
4.4 Palvelujen käyttöroolit

AE-entiteetillä on aina verkossa rooli. Rooli on aina joko palveluluokan tarjoaja (Service Class Provider, SCP) tai palveluluokan käyttäjä (Service Class User, SCU). SCP ja SCU AE -entiteetit keskustelevat toistensa kanssa verkon yli. Keskustelun aloittaa SCU, joka pyytää SCP-entiteetille yhteyttä avattavaksi, jonka jälkeen SCP joko hylkää tai hyväksyy yhteyden. Kun yhteys on auki, voidaan neuvotella erilaisista säännöistä, kuten esimerkiksi siitä, mitä binaaridatan tavujärjestystä yhteydessä käytetään, ja että sopiiko lähettäjän kuvanpakkaustapa vastaanottajalle. Säännöistä

sopimisen jälkeen voidaan neuvotella siitä, mitä toimintoja SCU haluaa SCP-entiteetiltä käyttää. SCU voi pyytää SCP-entiteetiltä esimerkiksi kuvan lähettämistä UID-tunnisteen avulla. Varsinaisesta neuvottelusta kerrotaan lisää seuraava kappaleessa.

4.5 DIMSE-palvelut

DIMSE-palvelun kutsuja on aina SCU (palvelun käyttäjä) AE ja siihen vastaa aina SCP (palvelun tarjoaja) AE. DIMSE-palvelussa SCU pyytää SCP-entiteetiltä jotain toimintoa. Tyypillinen toiminto voi olla esimerkiksi sellainen, että käyttäjä pyytää arkistosta tietoja potilaan talletetuista tutkimuksista C-FIND -palvelun avulla. C-FIND -palvelun jälkeen käyttäjä pyytää kyseisen tutkimuksen kuvat katsottavaksi C-GET -palvelulla. Kuvassa 6 on kuvattu kyseinen tapaus sekvenssikaaviona.



Kuva 6. Peräkkäiset C-FIND ja C-GET -kutsut.

DIMSE-palveluihin kuuluu pyyntö ja vastaus. Palvelukutsun nimessä on aina joko "Rq", joka tarkoittaa pyyntöä tai "Rsp", joka tarkoittaa vastausta. Pyyntö myös lajitellaan Normalized (N) ja Composite (C) -tyyppien mukaan. Normalized tarkoittaa, että palvelun datalle on vastaavuus tosielämässä. Composite tarkoittaa, että palvelun data on kooste tosielämää vastaavasta ja vastaamattomasta datasta. Composite-tyyppiset palvelut ovat nimeltään DIMSE-C -palveluita ja Normalized-palvelut nimeltään DIMSE-N. Tieto datan tyypistä lisätään palvelun nimeen. Esimerkiksi MOVE-pyyntö (Rq) Composite (C) -tyyppisen datan siirtämiseksi on "C-MOVE-Rq".

Myös palvelut käyttävät aikaisemmassa kappaleessa kuvattua VR-koodausta ja tagikirjastoa. Jotta palvelutagit voidaan erottaa datatageista, kaikki palvelutagit on sijoitettu (0000)-ryhmään. IOD-dataoliot sijaitsevat (0008) ja sitä ylemmissä ryhmissä. Tieto siitä, onko palvelukutsussa

dataolio mukana, kerrotaan tagin Data Set Type (0000,0800) sisällä. Jos palvelu sisältää dataolion, se lähetetään Upper Layer -protokollalla välittömästi palveluolion jälkeen.

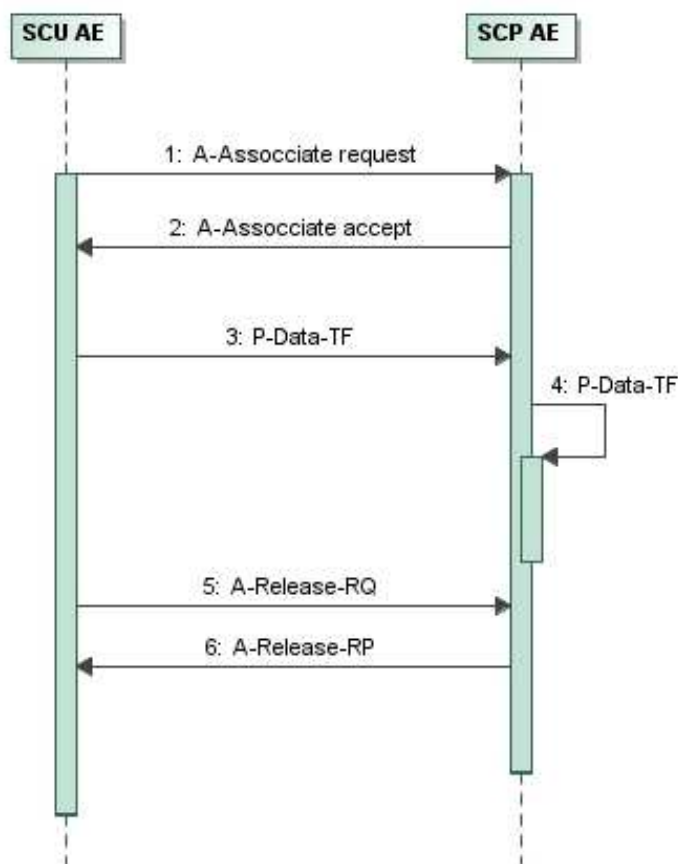
4.6 Upper Layer -protokolla (DICOM Associations)

DICOM määrittelee osiossa NEMA PS 3.8 [2011] DICOM Associations -protokollan, joka toimii TCP/IP-verkon ja DIMSE-palveluluokkien välissä (Katso kuva 4). Edellisissä kappaleissa kuvattujen IOD-dataolioiden ja DIMSE-palveluluokkien siirto toteutetaan tämän protokollan avulla. Protokollaa kutsutaan joissain tilanteissa nimellä DICOM UL, koska sen komennot toteutetaan TCP/IP-verkon välityksellä eli ”ylätasolla”.

DICOM UL yhdistää kaksi AE-entiteettiä yhteen, jotta ne voivat toteuttaa toistensa toiminnallisuuksia. Laitteet yhdistetään aina jokaisen transaktion alussa, jolloin ne sopivat yhteisistä säännöistä. Esimerkiksi silloin päätetään, käytetäänkö implisiittistä vai eksplisiittistä VR-koodausta.

On yleistä, että DICOM-verkon ylläpito kohtaa vaikeuksia juurikin tässä Associations-osuudessa. Ristiriitoja voi syntyä, koska yhteydenotto tapahtuu useassa kohdissa yksittäisten bittien tasolla. Kättelyn epäonnistuminen johtaa yleensä koko DICOM-verkon toimimattomuuteen, joten Association-osuuden epäonnistuminen huomataan yleensä varhaisessa vaiheessa. Vaikka Association-osuudessa on virhetilan välitysosio, eri ohjelmistovalmistajat yleensä jättävät virhekäsittelyn toteuttamatta [Pianyk, 2008].

Kuvan 7 UML sekvenssikaaviossa kuvataan se, kuinka Upper Layer -protokollan datan siirto toimii. Yhteys muodostetaan ensin A-Associate-RQ -komennon avulla. Yhteys hyväksytään A-Associate-Accept -komennolla (kuva 9) tai hylätään A-Associate-Reject -komennolla. Jos yhteys hyväksytään, dataa siirretään P-Data-TF -osiossa yksi dataosio kerrallaan. Tätä tehdään, kunnes kaikki data on saatu siirrettyä. Siirto voidaan myös peruuttaa SCP-entiteetin toimesta missä tahansa vaiheessa A-Abort -komennolla.



Kuva 7. UML sekvenssikaavio onnistuneesta tiedonsiirrosta kahden AE-entiteetin välillä.

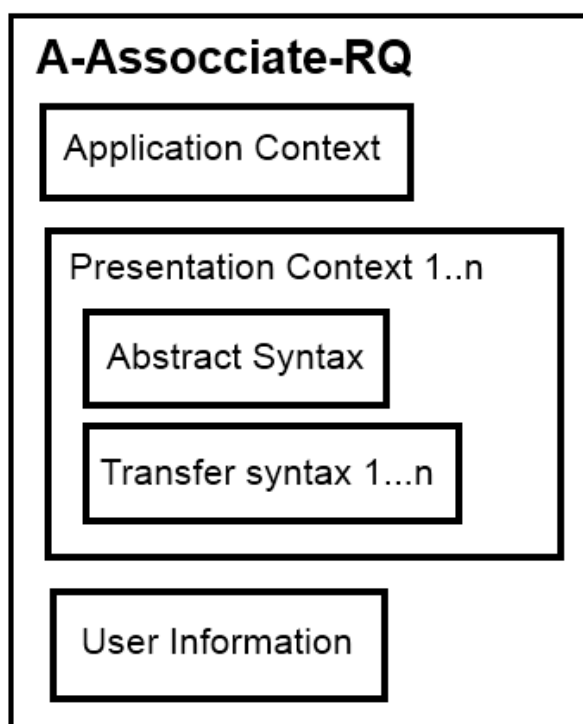
Jokaista pyyntöä DICOM UL -neuvottelussa kutsutaan nimellä Protocol Data Unit (PDU). Yllä kuvatun sekvenssikaavion PDU-pyyntöjen lisäksi on olemassa A-Abort sekä A-Negotiate-RJ PDU -pyynnöt, joista kerrotaan myöhemmin.

Seuraavaksi käydään läpi kaikki kuvan 7 vaiheet. Ensin kuvataan yhteysneuvottelu, jonka jälkeen kuvataan datan siirto ja mahdollinen keskeytys. Lopuksi kuvataan neuvottelun katkaisu.

4.7 Yhteysneuvottelu (Association Negotiation)

A-ASSOCIATE -neuvottelussa testataan se, ovatko AE-entiteetit keskenään yhteensopivia ja ovatko verkon asetukset sopivia. AE-entiteettien täytyy täyttää tiettyjä vaatimuksia, jotta ne voivat ymmärtää toisiaan. AE lähettää neuvottelussa kaikki tiedot itsestään. Vastaanottava osapuoli ilmoittaa, pystyykö se toimimaan kyseisen AE-entiteetin kanssa.

Neuvottelun aloittaja (SCU) lähettää ensimmäiseksi A-Associate-RQ -pyynnön vastaanottajalle. A-Associate-RQ -pyyntö sisältää kolme erityyppistä osiota, jotka on kuvattu kuvassa 8.



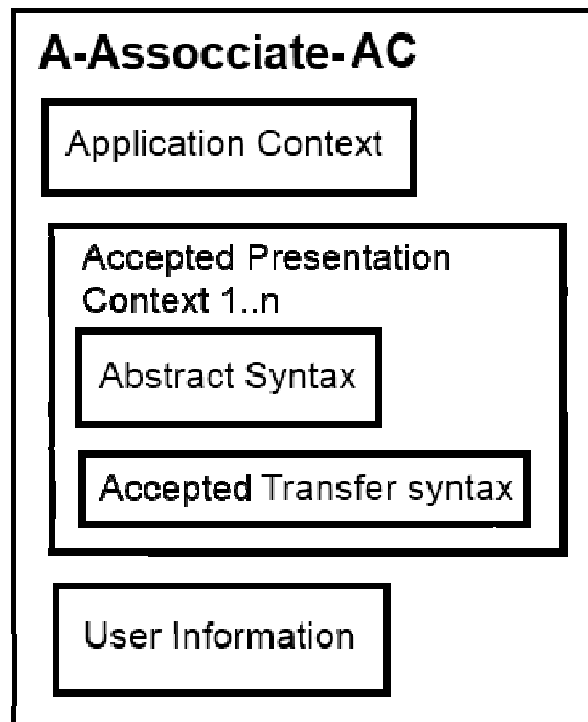
Kuva 8. Kaaviokuva A-Associate-RQ -sanomasta

Application Context -osio sisältää vain Application Context Name -parametrin. Kyseinen parametri sisältää käytännössä yleensä standardin tuetun version. Application Context jätetään usein huomioimatta.

Presentation Context -osioita voi olla useita. Vastaanottaja toimittaa A-Associate-AC -komennolla (kuva 9) ne Presentation Context -osiot, jotka se voi vastaanottaa. Abstract Syntax on tärkeä osa Presentation Context- osiota. Se sisältää SOP UID-koodattuna tiedon siitä SOP-luokasta, jota tässä pyynnössä halutaan käyttää. Erilaisia SOP UID-tyyppejä on esimerkiksi tulostamiseen, tietojen tallennukseen, tutkimustiedon noutamiseen sekä yhteyden tarkistamiseen (Verification). Eri SOP-luokkien toimintoja esitellään kohdassa 4.11. Transfer Syntax määrittelee sen, mikä on lähetettävän datan tavujärjestys sekä käytettävä kuvanpakkausalgoritmi. Transfer Syntax voi ehdottaa useita eri vaihtoehtoja ja AC-komennolla (kuva 9) kuitataan se Transfer Syntax, jota SCP haluaa käyttää.

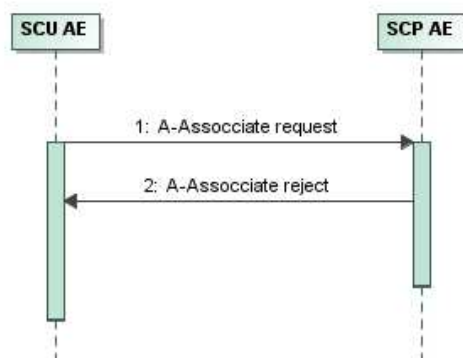
User Information -osio sisältää erilaisia tietoja pyynnön lähettäneestä AE-entiteetistä. Tärkeimmät tiedot User Information -osiossa ovat datan maksimipituus sekä kyseisen DICOM-ohjelmiston versio.

A-Associate-RQ -sanomaan vastataan aina joko A-Associate-AC- (kuva 9) tai A-Associate-RJ -sanomalla. A-Associate-AC -sanoma lähetetään, jos SCU-entiteetin parametrit sopivat SCU-entiteetille.



Kuva 9. Kaaviokuva A-Associate-AC -sanomasta

Jos vastaanottaja ei hyväksy ehtoja, niin se lähettää vastauksena A-Associate-RJ -sanoman, joka tarkoittaa yhteysneuvottelun hylkäämistä. Kuvassa 10 on kuvattu sekvenssikaavion avulla tilanne, jossa SCP hylkää yhteysneuvottelun.



Kuva 10. UML sekvenssikaavio siirron hylkäämisestä.

4.8 Datan siirto (P-Data-TF) ja yhteyden katkaisu

Kun yhteysneuvottelu on hyväksytty A-Associate-AC -sanomalla, siirrytään varsinaiseen datan siirtoon P-Data-TF -osion avulla. P-Data-TF eroaa muista PDU-komennoista, koska se sisältää DICOM-olioita. Lähetyksessä DIMSE-palveluluokat tulevat ensin ja sen jälkeen SOP-kutsuun liittyvä IOD-dataolio. Oliot pilkotaan P-Data-TF -siirrossa tietynkokoisiin palasiin. Näitä palasia kutsutaan PDV (Protocol Data Value) -osioiksi ja niitä voi olla yhdessä P-Data-TF -siirrossa useita. PDV-osion alussa ilmoitetaan niin sanottu PrC ID, joka on kättelyssä annettu edellisessä osiossa kuvatun Presentation Context -osion ID. Datan vastaanottaja pystyy PrC ID -tunnuksen perusteella tulkitsemaan, minkä tyyppinen SOP on kyseessä. Lisäksi sen avulla tiedetään, mitä kuvanpakkausta on käytetty ja että mikä on VR-dataelementeissä käytetty tavujärjestys. Presentation Contextin ID -osion jälkeen tulee PDV dataosio. Jokainen dataosio sisältää yhden tavun pituisen Message Control Header -otsikon ja varsinaisen VR-koodatun DICOM-datan. Message Control Header sisältää tiedon siitä, onko kyseessä DIMSE vai IOD sekä tiedon, onko kyseessä viimeinen olion palanen. Jos kyseessä on viimeinen palanen, vastaanottaja voi alkaa käsitellä kyseistä pyyntöä. Muussa tapauksessa vastaanottaja jää vielä odottamaan seuraavaa palasta. Koska P-Data-TF -siirrossa käsitellään monessa kohtaa dataa yhden bitin tarkkuudella, pieniinkään virheisiin olion datassa ei ole varaa. Muussa tapauksessa DICOM-sanomia ei pystytä tulkitsemaan vastapäässä.

Kun Message Control Header ilmoittaa, että kaikki data on siirretty, niin yhteys katkaistaan. Yhteyden katkaiseminen tapahtuu A-Release-RQ ja A-Release-RP -komentojen avulla. Yhteyden katkaiseminen tapahtuu aina, jos yhteyttä ei ole aikaisemmin katkaistu A-Abort tai A-Associate-RJ PDU -komennoilla. Joissain tilanteissa ohjelmistovalmistajat saattavat esimerkiksi lisenssisyistä rajoittaa yhteyksien määrää. Tällöin yhteys voi olla ohjelmoitu katkeamaan aikakatkaisun avulla, jos se ei jostain syystä ole katkennut [Pianykh, 2008].

Seuraavaksi kuvataan Finite State Process -prosessikaaviot ja sen jälkeen kuvataan DICOM UL -protokollan toiminta vielä niiden avulla.

4.9 Tutkielmassa käytetyt prosessikaaviot (FSP)

Tässä tutkielmassa kuvataan joitakin DICOM-datan tiedonsiirron eri tiloja *Concurrency: State Models & Java Programs, second edition* [Magee & Kramer, 2006] -kirjassa esitellyn Finite State Process (FSP) -notaation avulla. FSP perustuu automaatteihin, jotka ovat eri tilojen ja tilasiirtymien kuvaamiseen tehtyjä kieliä. Automaattien tarkoitus on kuvata järjestelmän kaikki mahdolliset samanaikaiset tilat ja toiminnot. Niiden avulla voidaan havainnollistaa järjestelmän käyttäytymistä. Automaatit ovat hyviä työkaluja erilaisten järjestelmien samanaikaisuuden suunnitteluun. FSP-kieli on suunniteltu erityisesti Java-ohjelmointikielen tarpeisiin.

Automaatteja voidaan havainnollistaa erilaisten diagrammien avulla. Tässä tutkielmassa havainnointikeinona käytetään Labeled Transition System (LTS) -tilakoneella muodostettuja kuvauksia (katso kuva 12). Magee sekä Kramer ovat myös olleet kehittämässä erillistä Labeled Transition System Analyzer (L TSA) -ohjelmistoa, jolla voi luoda LTS-kuvauksia FSP-kielen

kaavion perusteella. LTSA-ohjelmassa on analysointityökalut FSP-kielisen ohjelman oikeellisuuden tarkistamiseen. LTS-kuvaukset sopivat hyvin esimerkiksi yksinkertaisten DICOM-palveluiden esittämiseen. Sen sijaan jos tiloja on paljon, LTS-diagrammien antama informaatio yleensä kärsii. Tämän vuoksi esimerkiksi liitteen 1 koko FSP-kuvauksesta ei ole järkevää muodostaa LTS-diagrammia.

FSP-kielen prosessit ovat toiminnoista koostuvia kokonaisuuksia. Niiden sisällä on mahdollista suorittaa toimintoja erilaisten ehtojen mukaan. Prosessin sisällä voidaan myös kutsua toista prosessia tai erilaisia yksinkertaisia prosesseja. Yksinkertaisia prosesseja ovat esimerkiksi STOP eli ohjelman pysähtyminen ja virhe eli ERROR. Kuvassa 11 esitellään liitteen 1 FSP-määrittelyn mukaiset prosessit ROLES_SERVICES, NEGOTIATION ja TRANSFER_DATA. Prosessien sisällä valintaa kuvastaa merkki "|". Erilaiset tilat ja tilasiirtymät esitetään kielessä tilan kuvaksen ja "->" -symbolin avulla. Esimerkiksi kuvassa 11 tilat voitaisiin kuvata seuraavasti "c_echo_rsp -> associate_accept[role]".

FSP-kielen syntaksi koostuu yksinkertaisten prosessien lisäksi monimutkaisemmista eli niin sanotuista komposiittiprosesseista. Tässä tutkielmassa käytetään komposiittiprosessia "||", joka tarkoittaa rinnakkaisena tapahtuvaa toimintoa. Kyseisten rinnakkaisten toimintojen on tapahduttava kaikissa prosesseissa samanaikaisesti. Edellä kuvattujen toimintojen lisäksi FSP sisältää erilaisia operaattoreita, jotka noudattavat pääasiassa Java-kielen sääntöjä sekä vakioita, jotka määritellään yleensä FSP-kaavion alussa.

Kuvassa 12 on LTS-diagrammi Upper Layer -protokollan toiminnasta. LTS-diagrammissa tilat merkitään ympyröiden avulla. Ympyrän sisällä on numero, joka kuvaa sitä, missä järjestyksessä tilojen välillä siirrytään. Tilasta toiseen siirtyminen kuvataan viivan avulla, jossa on välissä nuoli kuvastamaan suuntaa. Yksittäisestä tilasta voidaan siirtyä useampaan tilaan ja tilaan voidaan päästä eri reittien kautta. Numerolla -1 merkitty tila tarkoittaa ERROR-tilaa, johon siirrytään virhetilanteissa. Siihen siirrytään joko virheellisessä määrittelyssä tai sitten kutsumalla FSP-kielessä ERROR-prosessia.

Tässä tutkielmassa kuvataan FSP-kielen ja LTS-diagrammien avulla DICOM UL -protokollan toimintaa ja sen jälkeen DIMSE-palvelut. FSP-kielen ja LTS-diagrammin avulla pyritään selventämään eri palveluiden toimintaa. Kappaleessa 6 kuvataan vielä DIM Web Service prosessikaavioiden avulla. Tutkielman ohessa toteutettu FSP-kuvaus löytyy kokonaisuudessaan tämän tutkimuksen liitteestä 1. Lisäksi liite 2 sisältää FSP-kuvauksen käännettulokset.

4.10 FSP-kaavio DICOM UL -protokollasta

Tässä osiossa kuvataan FSP-notaation avulla DICOM UL -protokollan toiminta. Alla oleva määrittely takaa sen, että tämän FSP-kuvauksen sisällä kaikkien DIMSE-kutsujen toiminnot synkronoidaan DICOM_UL -protokollan kanssa. DICOM_UL sisältyy kuvauksessa ROLES_SERVICES -prosessiin.

```
//DICOM_SERVICES = (ROLES_SERVICES||C_ECHO||C_MOVE||C_STORE||C_FIND||C_GET).
```

Kuva 11 sisältää FSP-prosessikaavion datan siirrosta. Aluksi kuvataan joitakin käytettäviä vakioita ja arvovälejä kuten AE-entiteettien roolit, joka on aina joko SCU tai SCP. Rooleissa SCP on 1 ja SCU on 0. Yhteyden vastapuoli saadaan laskettua saadaan FSP-kuvauksessa kaavalla $(role+1)\%2$. Eli jos $role=1$ niin kaavasta saadaan $role=0$ ja sama toisinpäin.

NEGOTIATION-prosessissa SCU-roolin omaava osapuoli lähettää ensin SCP-entiteetille pyynnön ja se joko hyväksytään tai hylätään. Varsinaista dataa siirretään TRANSFER_DATA - osiossa. RELEASE-toiminnalla päätetään yhteys.

```

const Request = 0
const Response = 1
const Cancel = 2
range Actions=Request..Cancel

const SCU = 0
const SCP = 1
range Roles=SCU..SCP

ROLES_SERVICES = (c_echo_rq -> NEGOTIATION[SCU]
|c_echo_rsp -> NEGOTIATION[SCP]
|c_store_rq -> NEGOTIATION[SCU]
|c_store_rsp -> NEGOTIATION[SCP]
|c_find_rq -> NEGOTIATION[SCU]
|c_cancel_find_rq -> NEGOTIATION[SCU]
|c_find_rsp -> NEGOTIATION[SCP]
|c_get_rq -> NEGOTIATION[SCU]
|c_get_rsp -> NEGOTIATION[SCP]
|c_cancel_get_rq -> NEGOTIATION[SCU]
),

NEGOTIATION[role:Roles] = (
associate_accept[(role+1)%2] -> TRANSFER_DATA[role]
|associate_reject[(role+1)%2] -> ROLES_SERVICES
|abort[(role+1)%2] -> ROLES_SERVICES
),

TRANSFER_DATA[role:Roles] = (
data_tf[role] -> RELEASE[role] | abort[(role+1)%2] -> ROLES_SERVICES),

```

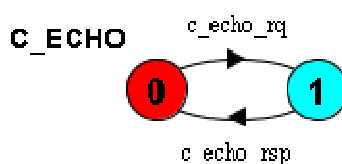

Verification SOP-luokalla tarkistetaan, että kaksi AE-entiteettiä on kytketty toisiinsa. Verification SOP käyttää DIMSE C-ECHO -palvelua, varsinaista IOD-dataa ei välitetä ollenkaan. C-ECHO on yksinkertainen DIMSE-palvelu. Se toimii niin, että SCU lähettää C-ECHO-Rq -kutsun, johon SCP vastaa C-ECHO-Rsp -kutsulla, kuten kuvassa 14 on esitetty. Jos vastausta ei saada, voidaan olettaa, että yhteyttä ei ole. Kuvassa 13 esitetään FSP-kielinen kuvaus C-ECHO DIMSE -palvelusta.

```

C_ECHO = C_ECHO[Request],
C_ECHO[action:Actions] =(
when (action==Request)
c_echo_rq -> C_ECHO[Response]
/when (action==Response)
c_echo_rsp -> C_ECHO).

```

Kuva 13. Prosessikaavio C-ECHO DIMSE -palvelusta.



Kuva 14. LTS-diagrammi C-ECHO DIMSE -palvelusta.

C-ECHO -palvelukomennot eivät sisällä IOD-dataoliota, joten niiden Data Set Type Parametrin arvo on ”ei dataa”. Vastauksen saanti kertoo, että C-ECHO -pyyntö oli onnistunut.

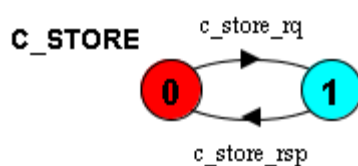
Storage SOP -palvelun avulla liikutetaan dataa entiteetistä toiseen DICOM-verkossa. Storage SOP -luokkia on olemassa useita. Kaikille erityyppisille kuvatyypeille, kuten magneettikuville ja ultraäänikuville on oma SOP-luokkansa. Eri kuvatyyppien SOP-luokkien toimintatapa on kuitenkin aina samanlainen. Storage SOP sisältää aina palvelun ja dataolion. Dataolio sisältää sen datan, jonka halutaan vastapään entiteetin tallentavan järjestelmäänsä. Storage SOP -luokat käyttävät C-STORE DIMSE -palveluluokkaa, kuten kuvassa 16 nähdään. Sanomanvälitykseen kuuluu aina pyyntö ja vastaus. Vastauksessa ilmoitetaan, että tallennettiin kuva onnistuneesti vastapuolen järjestelmään. Kuvassa 15 esitetään FSP-kielinen kuvaus C-STORE DIMSE -palvelusta.

```

C_STORE = C_STORE[Request],
C_STORE[action:Actions] =(
when (action==Request)
c_store_rq -> C_STORE[Response]
/when (action==Response)
c_store_rsp -> C_STORE).

```

Kuva 15. Prosessikaavio C-STORE DIMSE -palvelusta.



Kuva 16. LTS-diagrammi C- STORE DIMSE -palvelusta.

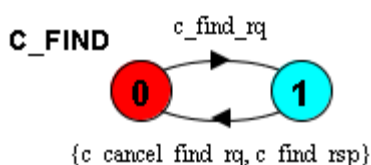
Query SOP -luokan avulla etsitään vastapuolen entiteettiin tallennettua dataa tietyin kriteerein. Haut voidaan kohdistaa käyttäjään, tutkimukseen tai niihin molempiin. Esimerkkihaku voisi olla vaikka yksittäisen päivän kaikkien tutkimusten haku. Käyttäjän asettamat hakukriteerit välitetään SOP-luokan IOD-dataolion sisällä. Vastauksena saadaan dataolion sisällä listan tutkimuksista tai potilaista. Näistä käyttäjä voi hakea haluamansa tutkimuksen tai potilaan. Varsinainen potilaan tutkimusten haku voidaan hoitaa seuraavaksi kuvattavalla C-GET SOP -palvelulla. Toisinaan pyynnön tekemisessä menee kauan aikaa. Tällaisessa tilanteessa pyynnön voi keskeyttää erityisellä C-Cancel -pyynnöllä, kuten kuvassa 18 osoitetaan. Kuvassa 17 kuvataan prosessikaavion avulla kysely C-FIND -pyynnöllä.

```

C_FIND = C_FIND[Request],
C_FIND[action:Actions] =(
when (action==Request)
c_find_rq -> C_FIND[Response]
/when (action==Response)
c_find_rsp -> C_FIND
/when (action==Response)
c_cancel_find_rq -> C_FIND
).

```

Kuva 17. Prosessikaavio C-FIND DIMSE -palvelusta.



Kuva 18. LTS-diagrammi C-FIND DIMSE -palvelusta.

Modality Worklist SOP käyttää C-FIND DIMSE -palvelua. Sitä käytetään kuitenkin erilaiseen tarkoitukseen. Modality Worklist -palvelun avulla käyttäjä voi noutaa SCP-osapuolelta kuvantamislaitteeseen päivittäisen kuvausaikataulun potilastietoineen. Potilaslistan haku tehdään yleensä kuvantamisprosessin alussa. Yleensä potilaslista noudetaan sairaalan RIS-järjestelmästä, jonne se tallennetaan tarvittaessa jostain muusta ajanvarausjärjestelmästä.

C-Get SOP -luokan avulla SCU AE noutaa SCP-osapuolelta joko yksittäisen tutkimuksen tai sitten potilaan kaikki tutkimukset. C-Get on aikaisemmin kuvatuista SOP-kuvauksia selkeästi monimutkaisin. C-Get -luokan käyttöön sisältyy C-Store SOP, jonka avulla siirretään varsinainen data. Alla kuvatun prosessikaavion (kuva 19) lisäksi useimmiten ennen C-Get -pyyntöä kutsutaan C-Find -palvelua, jolla haetaan yksittäisen tutkimuksen UID. C-GET-Rq -pyynnössä ilmoitetaan kyseinen haettu UID. Tämän jälkeen alkuperäinen SCP lähettää kyseisen tutkimuksen datan C-STORE-RQ -komennolla alkuperäiselle SCU-osapuolelle (kuva 20). C-GET-Rsp palauttaa tiedon siitä, onnistuiko tutkimuksen haku. Myös C-GET -luokassa voidaan käyttää C-Cancel -komentoa peruuttamaan haku, jos jostain syystä siihen on tarvetta.

```

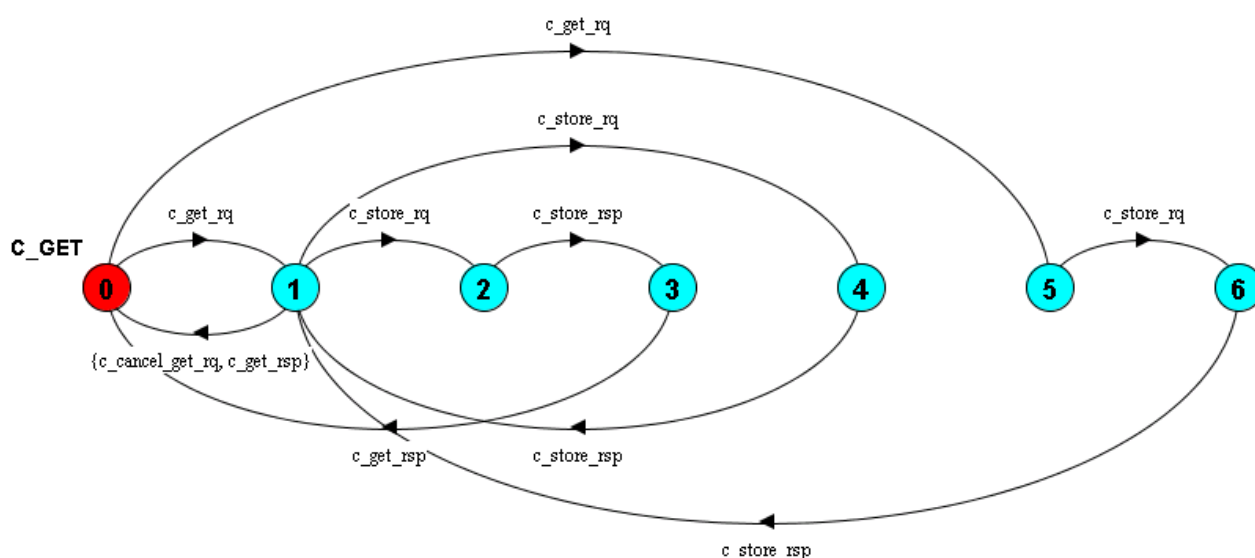
C_GET = C_GET[Request],
C_GET[action:Actions] =(
when (action==Request)
c_get_rq -> c_store_rq -> c_store_rsp -> C_GET[Response]
/when (action==Request)
c_get_rq -> C_GET[Response]
/when (action==Response)
c_store_rq -> c_store_rsp -> C_GET[Response]
  
```

```

/when (action==Response)
c_store_rq -> c_store_rsp -> c_get_rsp -> C_GET
/when (action==Response)
c_get_rsp -> C_GET
/when (action==Response)
c_cancel_get_rq -> C_GET
).

```

Kuva 19. Prosessikaavio C-GET DIMSE -palvelusta.



Kuva 20. LTS-diagrammi C-GET DIMSE -palvelusta.

Vielä monimutkaisemmaksi saattaa muodostua tilanne, jossa noudetaan kaikki potilaan tutkimukset. Tällöin haussa tehdään yksittäinen C-Get -kutsu, johon vastataan useilla C-STORE SOP -pyynnöillä, kuten kuvasta 20 nähdään. Jokaisen C-Store SOP -pyynnön jälkeen annetaan aina C-GET-Rsp -vastaus, jossa ilmoitetaan että kokonaishaku on vielä kesken. Kaikkien potilaan tutkimusten haut ovat kuitenkin harvinaisempia. Useimmiten haut kohdistuvat yksittäisiin tutkimuksiin ja C-Find -hakuun ennen C-Get -pyyntöä.

C-Move SOP vastaa C-Get SOP -luokkaa sillä erolla, että tutkimustietojen kohde, eli entiteetti johon data lähetetään, määritetään erikseen. Palvelu noudattaa prosessikaavion osalta C-Get -luokan toimintaa (kuva 20). C-Move -luokkaa voidaan käyttää myös C-Get -luokan sijaan sillä tavalla, että C-Move -luokan kohteeksi määritellään oma AE. Tämä on nykyään myös standardin suosittu tapa datan siirtämiseen. Osiossa NEMA PS 3.7 [2011] määritellään, että C-Move -luokkaa pitäisi käyttää C-Get -luokan sijasta, jos se on mahdollista. Lisäksi standardissa mainitaan, että C-Get on

standardissa vain taaksepäin yhteensopivuuden takaamiseksi. C-Move -luokan käyttö voi kuitenkin johtaa tietynlaisiin ongelmiin, josta kerrotaan lisää seuraavassa kappaleessa.

Storage Commitment SOP -palvelu vastaa Storage SOP -palvelua, mutta sen avulla voidaan varmistaa, että vastaanottaja varmasti arkistoi kuvat. Tämä palvelu on tavallaan enemmänkin sopimus kahden ohjelmiston välillä. Kyseinen palvelu on luotu sen takia, että joissain tilanteissa ohjelmiston logiikka on vastaanottanut kuvat, mutta säilyttää niitä vain väliaikaisesti. Kuvat säilytetään esimerkiksi vain kuvan tarkastelun ajaksi, mutta poistetaan välittömästi sen jälkeen. Talletuksen pysyvyys on kuitenkin hieman kyseenalaista. Vaikka Storage Commitment SOP -luokan avulla voidaan varmistaa kuvien arkistointi, vastapää ei voi kuitenkaan voida käytännössä luvata pysyvää säilytystä. Tutkimuksen tiedot voidaan joissain ongelmatilanteissa poistaa jälkikäteen joka tapauksessa. Yhtenä esimerkkinä on palvelimen levytilan loppuminen, jolloin yleensä on tapana poistaa kovalevyiltä vanhimpia tutkimuksia uusimpien säilyttämiseksi.

Secondary Capture SOP on tarkoitettu muiden kuin lääketieteellisten kuvantamislaitteiden kuvien tallentamiseen. Sen avulla voidaan esimerkiksi tallentaa kuvanmuokkauksia erilaisista päätteistä tai kameroista. Tätä SOP-luokkaa suositellaan käytettäväksi aina kun DICOM-verkossa liikutetaan dataa, jota ei ole luotu lääketieteellisellä kuvantamislaitteella.

Structured Reporting SOP on palvelu, jonka avulla voi välittää käytännössä millaista rakenteista dataa tahansa, kuten esimerkiksi äänitiedostoja. Data voi koostua sisäkkäistä elementeistä ja se voi sisältää esimerkiksi diagnooseja, laskutustietoja ja niin edelleen [Mildenberger, Echelberg, Martin, 2002]. Ennen Structured Reporting -palvelua tutkimukseen ei ollut mahdollista lisätä rakenteista dataa. Tällöin teksti kovakoodattiin suoraan kuvadataan. Tällaisessa tilanteessa datan erottaminen kuvasta on kuitenkin jälkeinpäin lähes mahdotonta. Tätä tapaa käytetään kuitenkin nykyään vieläkin, esimerkiksi tilanteessa, jossa ohjelmistovalmistaja ei tue Structured Reporting -palvelua [Pianykh, 2008]. Structured Report -palvelun data ovat samalla tavalla DICOM VR -tietotyypeillä välitettyä dataa, kuten muutkin DICOM-dataelementit. Sen tagit löytyvät myös muiden elementtien tapaan DICOM-standardin sanakirjoista. Structured Report on noussut suosituksi palveluksi ja sitä tuetaan laajalti. Structured Report -palvelusta julkaistiin vuoden 2011 standardiversiossa XML-muunnos, josta kerrotaan lisää kappaleessa 6.

Tässä kappaleessa kuvattiin DICOM-verkon tietoliikenneominaisuudet. Kappaleessa kuvattiin SOP ja DIMSE sekä DICOM UL ja erilaiset DIMSE-palvelut ja niiden käyttötarkoitus. Lisäksi kuvattiin prosessikaavioiden avulla palveluiden ja DICOM UL -protokollan toimintaa. Tutkielmassa on nyt kuvattu DICOM-datan luonti sekä tietoliikenne. Seuraavaksi pohditaan, mitä ongelmia standardi sisältää. Sen jälkeen kuvataan, millaisia ratkaisuja ongelmatilanteisiin löytyy.

5. Standardin ongelmat

Tässä kappaleessa pohditaan standardin suurimpia ongelmia. Kappaleessa kuvataan ensin point-to-point -tietoliikennearkkitehtuuri ja sen ongelmat avoimessa verkossa. Lisäksi pohditaan suurimpia DICOM-verkon sisäisiä ongelmia. Myöhemmissä osissa käsitellään binaarikoodauksen aiheuttamia ongelmia. Lisäksi kappaleessa pohditaan 1980-luvulla kehitettyä standardia nykypäivän tiedonsiirtonopeuksien näkökulmasta. Kappaleessa yritetään analysoida sitä, millaisiin ratkaisuihin tiedonsiirtokapasiteettien kasvaessa pitäisi keskittyä.

5.1 Point-to-point -tietoliikenne

Viimeaikoina on tullut suosituksi tavaksi katsella radiologisia kuvia muuallakin kuin radiologiayksikön sisällä. Kuvia ja dataa siirretään esimerkiksi VPN-verkon sisällä eri järjestelmien välillä. Pyynnöt kohdistetaan usein PACS-arkistoon, josta data siirtyy kolmannen osapuolen ohjelmistoon. Tämän lisäksi dataa liikutetaan usein myös HTTP-protokollan avulla sairaalan ulkopuolisiin järjestelmiin [Kaldoudi, Karaiskakis,2006]. Uusimpana suurena haasteena DICOM-arkkitehtuurissa on luoda useiden sairaaloiden välisiä verkkoja, joiden välillä on tarkoituksena siirtää kuvantamisdataa [Ivetic & Dragan, 2009].

DICOM-standardin tietoliikenne perustui alun perin standardin itse määrittelemään point-to-point -protokollaan. Kyseinen protokolla oli tarkoitettu datan siirtämiseen kahden ohjelmiston välillä 50-napaisen kaapelin avulla. Nykyään kyseinen protokolla on vanhennettu ja standardissa on siirrytty TCP/IP-verkon päällä toimivaan tiedonsiirtoon. Standardin omat palveluluokat ja DICOM UL -protokolla kuitenkin edelleen perustuvat vanhaan point-to-point -tyyppiseen arkkitehtuuriin. Data on tarkoitettu siirrettäväksi paikasta A paikkaan B niin, että molemmat kohteet ovat tiedossa. Datan vastaanottaja määritetään IP-osoitteen, portin ja AE Titlen tarkkuudella.

Useissa DICOM-ohjelmistoissa IP-osoite ja AE Title usein määritellään vastapään asetuksiin. Joissain tilanteissa kyseiset tiedot on määriteltävä sallittuihin yhteyksiin (niin sanottu White List), tai muuten keskustelu laitteen kanssa ei onnistu. Avoimempaan verkkoon siirryttäessä DICOM-standardi muuttuu usein käytännössä käyttökelvottomaksi tällaisten ohjelmistojen kohdalla. Tämä johtuu siitä, että avoimessa verkossa point-to-point -tekniikan vaatimaa tietämystä datan vastaanottajasta ei enää pystytä takaamaan.

Point-to-point -rajoittuneisuuden takia onkin siirryttävä käytännössä aina johonkin muuhun kuin DICOM-standardin tarjoamiin ratkaisuihin, jos dataa halutaan siirtää DICOM-verkon ulkopuolelle. DICOM-standardi tarjoaa osiossa NEMA PS 3.18 [2011] WADO-pyyntöjä ratkaisuksi tiedonsiirto-ongelmiin HTTP-protokollan avulla. WADO on kuitenkin vain kuvaus pyynnöistä. Siinä varsinaiseen toteutukseen ei oteta kantaa. Varsinaista standardia tapaa DICOM-datan HTTP-siirtoon ei siis vielä ole. Ohjelmistovalmistajat ovatkin kehittäneet usein omia ratkaisujaan näihin HTTP-protokollan avulla siirrettäviin DICOM-tutkimuksiin. WADO-palvelusta ja muista mahdollisista ratkaisuista kerrotaan lisää seuraavassa kappaleessa.

5.2 Tietoliikenteen rajoittaminen DICOM-verkossa

Tietoliikenteen rajoittaminen vain tunnettuihin entiteetteihin on muodostunut normaaliksi käytännöksi sairaaloiden DICOM-verkoissa. Yhteyksien rajoittaminen johtuu laitevalmistajien ohjelmistojen tietoturva-asetuksista. Näillä asetuksilla yritetään varmistaa verkon ulkopuolisten mahdollisesti harmillisten entiteettien pääsy potilaan dataan. Tämä on kuitenkin hieman nurinkurista ajattelua. Esimerkiksi Oleg S. Pinykh [2008] kuvaa kirjassaan *Digital Imaging and Communications in Medicine (DICOM)-A Practical Introduction and Survival Guide* tätä toimintatapaa kyseenalaiseksi. Hänen mukaansa tietoturvan ei pitäisi perustua yhteyksien rajoittamiseen. Tietoturvassa pitäisi keskittyä sen sijaan DICOM-verkon suojaamiseen ulkopuoliselta uhalta. DICOM-verkko voitaisiin tehdä turvalliseksi esimerkiksi VPN-verkon avulla. Yhteyksien rajoittaminen herättääkin epäilyksiä siitä, että verkkoon liitettyjen ohjelmistojen tietoturva ei ole muuten kunnossa jos heidän täytyy rajoittaa entiteettien pääsyä ohjelmistoonsa.

DICOM-sisäverkkokin aiheuttaa toisinaan ongelmia. Ongelmia syntyy silloinkin, kun verkossa ei ole vaihtuvia IP-osoitteita ja uusia laitteita lisätään harvoin. Suurimman ongelman sisäverkossa aiheuttaa aikaisemmassa kappaleessa kuvattu C-Move -palvelukutsu. C-Move -palvelupyynnössä määritellään se, että mistä ja mihin dataa liikutetaan. Dataa voidaan pyytää siirrettäväksi johonkin ulkopuoliseen entiteettiin. Useat ohjelmistovalmistajat kuitenkin määrittävät luotetuiksi yhteyksiksi ainoastaan ne entiteetit, jotka ovat etukäteen määritelty mahdollisiksi kohteiksi. Kaikki sisäverkon entiteetit eivät ole siis luotettavuudestaan huolimatta automaattisesti mahdollisia siirtokohteita. C-Move -palvelua käyttämällä joudutaan entiteettien tiedot siis lisäämään toisinaan kahteen eri paikkaan, eli luotettaviksi yhteyksiksi ja siirtokohteiksi. Tämä aiheuttaa lisätöitä ja selvitettävää verkon ylläpidolle.

DICOM-standardi tekee edellä mainitun tilanteen yleiseksi DICOM-verkossa. Standardi määrittelee nykyään C-Move -palvelukutsun suositelluksi tavaksi siirtää dataa C-Get -kutsun sijaan. Ohjelmistovalmistajien käytännöt yhdessä standardin suositusten kanssa lisäävät jo ennestään rajoittavan point-to-point -arkkitehtuurin ongelmia. Tämä on johtanut tilanteeseen, jossa DICOM-muotoinen data rajoittuu entistä enemmän suljetun verkon sisälle. Tämä on hieman nurinkurista, koska tavoitteena on selkeästi löytää keinoja välittää dataa avoimemmin sen sijaan, että suljettaisiin verkko entistä pienemmälle joukolle ohjelmistoja.

Point-to-point -tekniikan arkkitehtuuri ei myöskään tue datan edelleenlähetystä. Point-to-point -arkkitehtuurin idea on, että data määritellään siirrettäväksi paikasta A paikkaan B eikä siitä eteenpäin. Datan kohde on siis aina sen päätepiste. Jos dataa halutaan edelleenlähettää, täytyy tehdä jokin standardin vastainen tapa esimerkiksi luoda DIMSE-palvelu uudelleen ja lähettää data esimerkiksi C-Store SOP -luokan avulla. Tämä tuo lisätyötä ohjelmistosuunnittelijalle ja mahdollisesti lisäkustannuksia sairaalalle, koska kyseistä toimintoa ei standardin puolesta toteuteta.

5.3 Binaaritavuihin perustuva koodauksen edut ja haitat

DICOM-tietoliikenteen ongelmien lisäksi monesti kritisoidaan DICOM-datan binaaritavujen avulla koodattua tagirakennetta, joka koetaan hankalaksi ja epäintuitiiviseksi. Tässä osiossa kerrotaan hieman suurimmista ongelmista DICOMin binaaridataelementein koodatussa datassa.

DICOM-datan rakenne perustuu niin sanottuihin VR-elementteihin ja tagikirjastoon. Elementtien sisältö ja datatyypin kuvaus ilmoitetaan ennen dataa usein yksittäisten bittien ja tavujen tasolla. Koska tagit määritellään standardin puolesta erillisessä kirjastossa, niitä ei toisteta uudelleen varsinaisessa datassa. Molemmilla osapuolilla on tiedossa sama tagikirjasto, jotta he voivat tulkita dataa samalla tavalla. Kyseinen koodaus voidaan lukea eduksi, koska se säästää useita tavuja jokaista elementtiä kohden. Verkot ovat kuitenkin nopeutuneet huomattavasti nykypäivänä ja 1980-luvulla suunniteltu elementtirakenne on väistämättä vanhentunut. Muutamien tavujen säästämällä ei nykyisillä tiedonsiirtokapasiteeteilla ole enää niin suurta merkitystä.

Haasteita kuitenkin myös verkkoliikenteen nopeuden osalta löytyy edelleen suurissa määrissä. Näyttöjen parantuessa ja kuvien kuvapistemäärän suuretessa siirrettävä datamäärä kasvaa jatkuvasti. Kuvadatan pakkaus onkin nykyään kasvanut suureksi haasteeksi nopeamman tiedonsiirron saavuttamiseksi. Erilaisia tehokkaita kuvanpakkausalgoritmeja käyttämällä säästettävä tiedonsiirtoaika on usein moninkertainen verrattuna DICOM BMP -muotoisen datan siirtämiseen. Voidaankin sanoa, että haasteet ovat muuttuneet 1980-luvulta nykypäivään verrattuna eri mittakaavan ongelmiin.

5.4 Monimutkaisia tietotyyppisiä, määrittelyitä ja elementtejä

Jotkut standardin määrittelemät tietotyypit ovat osoittautuneet ongelmallisiksi. Historiallisen taustansa perusteella niitä on vaikea muuttaa jälkeenpäin. Joistain yleisistä päivää ja aikaa kuvaavista attribuuteista löytyy esimerkiksi joitain epäjohdonmukaisuuksia. Potilaan syntymäaika (Patient's Birth Time) ja potilaan syntymäpäivä (Patient's Birth Date) -attribuutit ovat hyvin yleisluontoista tietoa potilaasta. Ne ilmoitetaan DICOM-datassa aina eri kentissä. Tämä tarkoittaa, että DICOM-datan päivityksessä molempia kenttiä täytyy muuttaa. Loogisempaa olisi ollut alun perin käyttää pelkkää yksittäistä kenttää syntymähetken ilmoittamiseen. Tässä tilanteessa olisi ollut hyvä käyttää standardin määrittelemää VR Datetime (DT) -tietotyyppiä kuvaamaan näitä tietoja yhdessä kentässä. Tämä saattaa vaikuttaa pieneltä ongelmalta, mutta tämäläpaisia ongelmakohtia löytyy useita. Yksittäisiä tietoja päivitetäessä koko DICOM-data joudutaan tekemään uusiksi, jos datan pituus muuttuu. Tämä johtuu siitä, että dataosion pituus ilmoitetaan esimerkiksi DICOM-olioissa ja myös koko dokumentin alussa.

Historian aikana joitain tietotyyppisiä on silti muokattu. Esimerkiksi aikaisemmissa DICOM ja ACR-NEMA -versioissa aikamuoto oli HH:MM:SS ja päivämäärä muodossa DD.MM.YYYY, mutta ne ovat muuttuneet muotoon HHMMSS ja DDMMYYYY. Tämä vaikeuttaa hieman datan lukua vanhojen kuvantamislaitteiden kohdalla, koska ne saattavat noudattaa edelleen vanhoja

standardiversioita. Taaksepäin yhteensopivuus onkin tässä kohtaa otettava huomioon ympäristöissä, joissa on käytössä vanhoja laitteita.

Yksi erikoinen tietotyyppi on UN VR eli Unknown-tietotyyppi. UN VR kuvaa sellaista datatyyppiä, jota ei standardin puolesta löydy VR-tietotyyppien listalta. Tämä on useimmiten ongelmallinen tietotyyppi ja sitä kehoitetaan välttämään mahdollisimman paljon [Pianykh 2009]. UN VR -tietotyyppissä ongelmana on tavujärjestys, jota ei voida määritellä etukäteen. Jokaisesta UN-tietotyyppiä käyttävästä dataelementistä pitääkin sopia kaikkien dataa vastaanottavien osapuolten kesken. Tavujärjestys ja sisältö tulee olla tiedossa, jos dataa halutaan jotenkin tulkita vastapään toimesta.

Jotkut lähteet ihmettelevät sitä, että käytännössä DICOM-ohjelmistot eivät useinkaan noudata standardin määrittelemiä sääntöjä esimerkiksi joidenkin VR-elementtien kohdalla. Yhtenä selkeimpänä esimerkkinä pidetään aikaisemminkin mainittua nimen koodausta, jota ohjelmistovalmistajat eivät useinkaan noudata.

5.5 Parillisuussääntö

Kuten aikaisemmin todettiin, DICOM-elementin arvon pituuden tulee aina olla parillinen. Jos näin ei ole, arvoon lisätään yksi merkki, jotta siitä saadaan parillinen. Tämä on ehkä yksi erikoisimmista ratkaisuista, joihin DICOM-standardissa on päädytty. Tälle on kuitenkin joitakin käyttötarkoituksia. Yksi yleinen käytötapa tälle on DICOM-datan oikeellisuuden tarkastaminen. DICOM-data voidaan hyvin yksinkertaisesti tulkita korruptoituneeksi, jos se ei sisällä paritonta määrää merkkejä. Virhetarkistus kuitenkin olisi järkevämpi tehdä esimerkiksi datan CRC-tarkistuksella. Dataelementin tai koko DICOM-olion CRC-tarkistus olisi luottavampi tapa saada tieto datan virheestä.

Parillinen datakenttä synnyttää enemmän ongelmia, kuin mikä siitä saatu hyöty on. Suurin ongelma lienee datan lukeminen järjestelmien välillä. Kenttien dataa ei voida lukea sellaisenaan vastaanottavaan järjestelmään, vaan tyhjat merkit on poistettava aina datan lopusta. Tämä on sinänsä helppo operaatio, mutta herättää kysymyksen siitä, olisiko ollut järkevämpää alun perin suunnitella elementit niin, että ne sisältäisivät juuri sen datan, joka halutaan vastapuolelle välitettävän.

5.6 DICOM-datan muunnostarpeet

Oleg S. Pianykh [2008] mainitsee joitakin esimerkkejä käytännön tilanteista, joissa standardia ei noudateta. Esimerkiksi potilas-id -tiedon (Suomessa henkilötunnus) mukana saattaa olla merkkejä, jotka esimerkiksi viittaavat tutkimuksen ottajan nimeen. Ongelma lienee osittain siinä, kuinka DICOM sulautuu terveydenhuollon prosesseihin. Jos potilas-id haettaisiin aina esimerkiksi perusjärjestelmästä, tällaista ongelmaa ei pääsisi syntymään. Modality Worklist SOP on tarkoitettu noutamaan potilas-id RIS-järjestelmästä. Syynä siihen, miksi kyseistä toimintoa ei käytetä voi olla RIS-järjestelmien integraatioiden puuttuminen. RIS tarvitsee ajanvaraustiedot yleensä jostain

ulkoisesta ajanvarausjärjestelmästä ja ajanvarausten synkronointi on määritetty standardin puolesta toteutettavaksi HL7-sanomaliikenteellä. Tämä kuitenkin vaikuttaa hieman monimutkaiselta tavalta.

DICOM-standardi on selkeästi joissain tilanteissa liian monimutkainen ja vaikeaselkoinen. Tämä nähdään jo siitä, että usein DICOM-ohjelmistotkaan eivät noudata sitä, kuten joissain aikaisemmissa tapauksissa on kuvattu. Nykyään onkin tullut yhä suositummaksi muuttaa DICOM-data helpokäyttöisempään ja yleiseen XML-muotoon erilaisilla työkaluilla. Useimmat ohjelmointikielet tukevat sellaisenaan XML-muotoisen datan lukua oliorakenteeksi. Lisäksi erilaisia työkaluja datan muunnokseen löytyy suuret määrät. Esimerkiksi Jumppanen, Jaatinen, Toivanen [2005] esittelevät artikkelissaan valmiin tagikirjaston XML-muunnokseen.

Lähes kaikki tässä kappaleessa kuvatut DICOM-standardiin liittyvät ongelmat voidaan todennäköisesti ratkaista XML-muunnoksen sekä erilaisten HTTP-pyyntöihin perustuvien ratkaisujen avulla. DICOM-verkot ja DICOM-muotoinen data tulee kuitenkin pysymään jatkossakin käytössä sairaaloiden sisällä. Tämä johtuu siitä, että vanhat kuvantamislaitteet on suunniteltu tukemaan vain DICOM-standardin versio 3.0 -muotoista dataa. On todennäköisesti liian kallista ohjelmoida laitteet uudelleen niin, että ne tukisivat XML-muotoa. Muualla kuin DICOM-verkon sisällä on kuitenkin suositeltavaa käyttää muuta kuin DICOM VR -enkoodattua dataa. XML-muunnosta ja HTTP-protokollan kautta välitettävää dataa käsitellään lisää seuraavassa kappaleessa.

6. Vaihtoehtoisia lähestymistapoja

Edellisessä kappaleessa kuvattiin DICOM-standardiin liittyviä ongelmia. Niitä ovat esimerkiksi point-to-point -tekniikka ja DICOM-muotoisen datan vaikeaselkoisuus. Tässä kappaleessa yritetään etsiä ratkaisuja kyseisiin ongelmiin. Kappaleessa kuvataan tapoja, kuinka DICOM-muotoista dataa voidaan välittää eri järjestelmien välillä ilman point-to-point -tyyppisen tekniikan rajoituksia. Lisäksi kuvataan, kuinka DICOM-data voidaan esittää selkeämmässä muodossa. Ratkaisut tässä kappaleessa keskittyvät XML-muotoiseen dataan. Datan siirto kuvataan HTTP-pyyntöjen avulla ja HTTP-protokollan kautta välitettävillä SOAP-kutsuilla.

6.1 DICOM datan ja XML-tiedon rakenne

Tämän tutkielman kappaleessa 3.6 kuvattiin DICOM-datan avulla potilaan nimi. Kyseisessä esimerkissä nähtiin, kuinka elementtien rakenne perustuu DICOM-datassa osittain binaaritavujen avulla tehtävään koodaukseen. Kuten osiossa kerrottiin, datan kaikki mahdolliset tietosisällöt määritellään erikseen standardin puolesta. Standardi kuvaa kaikki mahdolliset tagit, joita dataoliassa voidaan käyttää.

XML on metakieli, jonka avulla voidaan kuvata myös DICOM-dokumentin rakenne. XML on tapa esittää monimutkaista dataa ihmisen luettavassa muodossa. XML on suosittu esimerkiksi Internetissä, jossa HTML-kieli perustuu XML-kieleen. DICOM ja XML koostuvat molemmat sisäkkäisistä elementeistä, joten niitä voidaan pitää samankaltaisina tietorakenteina. XML kuvaa sisäkkäiset elementit aloitus ja lopetustageilla, joissa tietue kuvataan merkkijonona < ja > -merkkien välissä. Esimerkiksi potilaan nimi kuvataan NEMA PS 3.20 [2011]-osion XML-kuvauksessa näin:

```
<name>
  <given>Matti</given>
  <family>Meikäläinen</family>
</name>
```

Tästä XML-kuvauksesta voidaan jo nähdä, mikä on XML-muotoisen datan etu verrattuna osiossa 3.6 esitettyyn binaarimuotoiseen DICOM-elementtirakenteeseen. DICOM-koodatuissa elementeissä on ilman tagikirjastoa mahdotonta olla varma siitä, mitä data sisältää. Sen sijaan XML-datasta voi päätellä sisällön tyyppin ilman ylimääräisiä tagikirjastoja.

XML on yleistynyt nykyään useimpien ohjelmointikielten työkaluihin. Sen takia voidaankin ajatella, että DICOM-dataa käsittelevissä ohjelmistoprojektissa on huomattavasti helpompaa muuttaa data ennen käsittelyä XML-muotoon. Useimmiten on työläämpää toteuttaa DICOM VR -koodattua dataa lukeva erillisohjelmisto [Kaldoudi, Karaiskakis,2006].

6.2 XML-muunnokset ja vuoden 2011 HL7-XML -muunnos

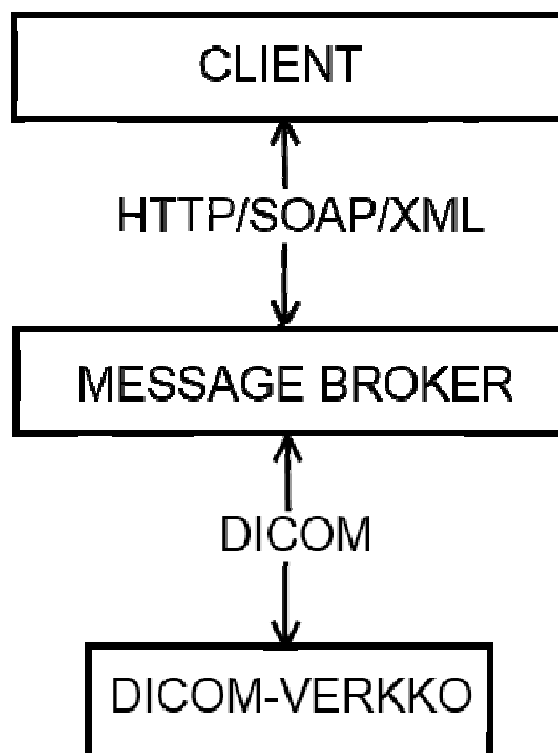
Standardin osio NEMA PS 3.20 [2011] esittelee DICOM-XML -muunnostyökalun HL7 (Health Level 7) -määrittelemän sanomanvälitysstandardin versio 3 muotoon. HL7 versio 3 on kehittynyt versiosta 2 niin, että sanomat ovat nykyään XML-muotoa. Versiossa 2 sanomat olivat erilliseen standardin määrittelemään muotoon koodattua merkkijonoja. Vaikka version 2 sanomat ovat esimerkiksi Suomessa vielä laajalti käytössä, niin myös suomalaisten tietojärjestelmien integraatioissa ollaan alkamassa suosia version 3 tyyppistä tiedonvälitystä.

Koska tarpeet XML-kuvaukseen ovat olleet olemassa jo pitkän aikaa, olemassa olevien muunnostyökalujen kirjo on nykyään laaja. Yksi suosittu ohjelmisto muunnoksen tekemiseen on JDCM-kirjasto, jonka dcm2xml-työkalun avulla voi tehdä DICOM-XML -muunnoksen. JDCM sisältää myös työkalun, jolla XML-muotoisen datan voi muuntaa takaisin DICOM-muotoon [Nagy, 2007].

Koska NEMA PS 3.20 [2011] esittelee XML-muunnoksen vasta lähes kolmekymmentä vuotta standardin kehityksen jälkeen, tilanne voidaan ajatella hieman samankaltaiseksi kuin standardin alkuvaiheessa. Alkuun ohjelmistovalmistajilla oli ohjelmistokohtaisia tapoja siirtää dataa ohjelmistojen välillä. Nyt ongelmana on eri ohjelmistovalmistajien DICOM-XML -käännökset, joita löytyy runsaasti. Tilanne lieneekin sekä standardin, että ohjelmistovalmistajien kannalta hieman ongelmallinen. Ohjelmistovalmistajat joutuvat joko standardoimaan XML-muunnoksensa tai sitten käyttämään standardin vastaista tapaa kuvata DICOM-dattaa. Tulevien integraatioiden osalta NEMA PS 3.20 [2011] todennäköisesti kuitenkin tuo helpotuksia. Kyseisessä osiossa muunnoksessa käytetty HL7 versio 3 on standardi muoto, jota suositetaan sairaalajärjestelmien välisissä integraatioissa.

6.3 Datan siirto HTTP-protokollan avulla

Useissa tutkielmissa mainitaan tapa avoimeen DICOM-datan välittämiseen HTTP-pyyntöjen avulla. HTTP-pyyntöjen vastaanottaja on useimmissa tapauksessa DICOM-verkossa sijaitseva palvelin tai sanomien muuntaja (Broker). Broker voi muuntaa HTTP-pyyntöt DICOM-pyyntöiksi ja palauttaa vastauksen esimerkiksi HTTP-pyyntönsä rungossa XML-tyyppisenä datana. Kuvassa 21 on kuvattu tämäntapainen muunnosohjelmisto (Message Broker).



Kuva 21 Broker muuttaa DICOM-verkon sanomat tarvittavaan muotoon

Tässä tutkielmassa esitellään kuvan 21-tyyppisistä muunnoksista DICOM-standardin PS 3.18. [2011]-osion WADO sekä artikkelissa *A service based approach for medical image distribution in healthcare Intranets* kuvattu DIM Web Service [Kaldoudi, Karaiskakis,2006]. Web Service -palveluista on tullut entistä yleisempi tapa välittää dataa eri järjestelmien välillä. Myös DICOM-standardi julkisti vuoden 2011 versiossa PS 3.19 [2011]-osiossa Web Service -ratkaisun DICOM-palvelupyyntöjen muuntamiseen. Kyseinen PS 3.19 [2011]-osion Application Hosting on kuitenkin hieman erilaiseen tilanteeseen tarkoitettu ratkaisu, kuin voitaisiin odottaa. Sen sijaan, että siinä olisi kuvattu WADO-tyyppinen sanomien muunnostyökalu, kyseessä on ratkaisu, jolla toinen ohjelmisto voidaan liittää palvelinjärjestelmään plugin-tyyppiseksi lisäosaksi. DICOM-standardin vuoden 2011 perusteella NEMA kuitenkin on myös omalta osaltaan alkanut uudistaa standardia avoimemman verkkoliikenteen puolesta.

6.4 Kuvaus Web Service -palveluista ja käytöstä DICOM-datan välityksessä

Web Service- palvelut ovat WWW-verkossa sijaitsevia moduuliohjelmia. Web Service -palveluissa sanomanvälityksen hoitaa Simple Object Access Protocol (SOAP), joka toimii HTTP-protokollan yläpuolella. Web Service -palvelut perustuvat XML-muotoiseen dataan. Myös SOAP-pyyntöt ovat

XML-muotoisia. Web Service -palveluiden käyttöönotto tapahtuu niiden WSDL (Web Service Description Language) -kuvausten avulla. WSDL-kuvaus sisältää funktiot ja tietokantaoliot, joita vastapuoli voi ottaa käyttöönsä. Universal Description, Discovery and Integration (UDDI) julkaisee rekisterissään tietokantaa erilaisista Web Service -palveluista.

Web Service -palvelut ovat Service-Oriented Architectures (SOA) -tyyppisiä ratkaisuja. SOA tarkoittaa kokoelmaa palveluita, jolla voi välittää erilaisia sanomia verkon välityksellä. SOA-tyyppisiä ratkaisuja on ollut käytössä jo ennen Web Service -palveluita. Yleisesti käytössä olevia SOA-ratkaisuja ovat Object Management Group -yhdistyksen määrittelemä CORBA (Common Object Request Broker) ja Javan etäoliokutsuihin tarkoitettu RMI (Remote Method Invocation). Web Service -palveluiden yhtenä etuna pidetään riippumattomuutta ohjelmointikielestä, koska kieli on kokonaan XML/SOAP-muotoista. Web service -palvelut ovat yleistyneet viime vuosina varsinkin HTTP-protokollan laajan käytön ja Internetin tietoliikenteen lisääntyessä. Erilaisissa WWW-palveluissa on lisääntyvä tarve siirtää dataa järjestelmästä toiseen verkon yli. Web Service -palveluiden etuna pidetään sitä, että vastapään järjestelmän ei tarvitse tietää toteutuksesta. He tarvitsevat pelkän XML/SOAP-rajapinnan sanomien välittämiseen [Kaldoudi, Karaiskakis,2006].

Erilaisia DICOM-Web service -ratkaisuja on vuosien mittaan kehitelty eri tahojen toimesta. Monissa tieteellisissä artikkeleissa on annettu oma toteutus XML-muotoisen DICOM-datan integroimiseen [Jensen, Baumgartner, 2003], [Prinz, Fischer, Schuster, 2005] [Kaldoudi, Karaiskakis,2006]. Artikkelissa *A service based approach for medical image distribution in healthcare Intranets* Kaldoudi ja Karaiskakis [2006] käsittelevät sitä, kuinka DICOM-data voidaan integroida järjestelmien välillä DIM-nimisellä Web Service -ratkaisulla. Kyseinen ratkaisu kuvataan myöhemmin tässä kappaleessa.

Myös uusin versio DICOM-standardista (2011) kuvaa osiossa NEMA PS 3.19 [2011] Application Hosting Web Service -rajapinnan. Kyseisessä ratkaisussa palvelinohjelmistoon (Hosting System) liitetään toinen plugin-tyyppinen lisäosa (Hosted Application). Osiossa kuvataan, kuinka ohjelmistojen välinen tiedonsiirto tapahtuu SOAP-kutsuilla. Osio sisältää myös WSDL-kuvaukset pyyntöjen tekemiseen. Application Hosting kuvataan tarkemmin myöhemmin tässä kappaleessa.

6.5 HTTP-protokolla ja transaktioiden ongelmat

Transaktiot koetaan usein ongelmallisina HTTP-pyyntöissä. Transaktioita voidaan pitää myös Web Service -palveluiden ongelmana. Vastapään tietokannan tila on usein vaikea tulkita ennen pyyntöä. Vastapään tietokannan lukitseminen transaktion ajaksi on myös yleensä kyseenalaista. Voidaankin todeta, että tietokannan eheys saattaa HTTP-pyyntöjä käyttämällä heikentyä.

Transaktioiden toimivuus ja käyttötavat riippuvat HTTP-liikenteessä usein palvelun käyttäjämääristä. Jos käyttäjäkunta sisältää vain muutamia käyttäjiä, voi järjestelmän kannalta olla perusteltua lukita käyttäjän muokkaamat tietokantarivit aina HTTP-pyyntöjen välillä. Sen sijaan jos käyttäjiä on satoja, lukitseminen saattaa jumittaa koko järjestelmän.

Transaktioiden suurin ongelma on se, että tietokannan tila saattaa olla muuttunut käyttäjän muokkauksen ja lomakkeen lähetyksen välillä. Tämä saattaa johtaa ongelmiin, johon käyttöliittymän olisi hyvä ottaa kantaa jotenkin. Ratkaisu voisi olla esimerkiksi, että käyttäjälle ilmoitetaan arvojen muuttumisesta ennen tallentamista. Tämän jälkeen käyttäjältä voidaan kysyä, ovatko uudet arvot sopivia tietokannan tilaan nähden.

Lukitseminen ja transaktioiden sujuvuus riippuu myös siitä, onko järjestelmä suunniteltu ottamaan tietokantaan yhteys jokaisen käyttäjän osalta. Jos tietokantayhteyksiä on rajattu määrä, niin täytyy miettiä esimerkiksi sitä, asetetaanko pyynnöt jonkinlaiseen jonoon odottamaan tietokannan vapautumista.

Myös DICOM-dataa käyttävää palvelinta suunnitellessa on tarpeen miettiä transaktioiden sujuvuutta. Transaktioiden käyttöä voidaan esimerkiksi miettiä siltä kannalta, montako samanaikaista käyttäjää palvelulla mahdollisesti on. Käyttäjiä voi olla muutamasta käyttäjästä useisiin satoihin käyttäjiin riippuen palvelun tyypistä. Mitä enemmän käyttäjiä on, sitä tärkeämmäksi transaktioiden suunnittelu muuttuu. DICOM Web Service -ratkaisuja käytettäessä transaktioiden kannalta eniten suunnittelua vaativa tilanne on tutkimusdatan tallentaminen. Tällöin on tärkeitä keskittyä siihen, että tietokanta on eheä ennen tallennusta sekä sen jälkeen.

6.6 WADO

DICOM määrittelee osiossa NEMA PS 3.18 [2011] HTTP-pyyntöjen avulla tehtävät pyynnöt ja DICOM-datan välityksen WADO-palvelun avulla. Alla on esimerkki WADO-muotoisesta HTTP-pyyntöstä, jossa vastapäättää pyydetään palauttamaan kuva lähettäjälle. Pyyntössä välitetään halutun tutkimuksen UID-tunnus.

```
http://www.hospital-stmarco/radiology/wado.php?requestType=WADO
&studyUID=1.2.250.1.59.40211.12345678.678910
&seriesUID=1.2.250.1.59.40211.789001276.14556172.67789
&objectUID=1.2.250.1.59.40211.2678810.87991027.899772.2
```

WADO-palvelua on kritisoitu monessa eri artikkelissa siitä, että sen pyynnöt on määritelty hyvin löysästi ja niistä puuttuu tiettyjä tärkeitä DICOM-palveluita [Kaldoudi, Karaiskakis, 2006], [Pianyk, 2008]. WADO ei esimerkiksi kuvaa sitä, mitä pyynnöille pitäisi tehdä ja millaista dataa pitäisi palauttaa. Kaldoudi, Karaiskakis [2006] pitävät suurimpana ongelmana sitä, että pyynnöt eivät esimerkiksi sisällä lainkaan hakuja (C-FIND). Tämä tekee mahdottomaksi WADO-palvelun kautta pelkän potilaan tiedoilla tehtävät tiedonhauut, koska tutkimuksen yksilöivä UID täytyy tietää aina etukäteen. WADO ei myöskään tue C-STORE -komentoa, joten sen avulla ei voida lisätä dataa palvelimelle.

Jos NEMA päättää laajentaa PS 3.18 [2011]-määrittelyä jossain vaiheessa hakuihin ja datan lisäyksiin, niin siitä saadaan mahdollisesti käyttökelpoinen työkalu. Lisäksi määrittelyssä olisi tärkeitä kuvata se, mitä pyynnöillä tarkalleen tehdään vastapäässä. Tällä hetkellä WADO on varsin

rajoittunut ratkaisu, eikä sen käytölle voi kovin helposti löytää perusteluita. Parempia ja laajempia ratkaisuja on olemassa, kuten esimerkiksi seuraavaksi kuvattava DIM Web Service.

6.7 DIM Web Service

Kaldoudi, Karaiskakis [2006] määrittelevät artikkelissaan *A service based approach for medical image distribution in healthcare Intranets* DIM-nimisen Web Service -ratkaisun. Ratkaisu muistuttaa hieman edellisessä kappaleessa kuvattua WADO-palvelua. DIM kuitenkin sisältää myös sellaisia tärkeitä toimintoja, jotka WADO-palvelusta puuttuvat.

DIM Web Service -ratkaisun komennot muistuttavat DICOM-palvelupyynnöjä. Esimerkiksi kuva pyydetään RetrieveImage-komennolla ja vastaus saadaan RetrieveImageResponse-komennolla. RetrieveImageResponse-komennossa kuvadata on <ImageData> -elementin sisällä. Kuva on koodattu tavu kerrallaan merkkijonomuotoon, esimerkiksi *FF D8 FF 00 10 4A*. Edellisessä osiossa esitellystä WADO-palvelusta puuttuu kokonaan kuvaus siitä, miten varsinainen kuva tulisi HTTP-pyynnössä välittää. DIM siis kuvaa pyyntöjen toiminnan huomattavasti WADO-palvelua tarkemmin.

Kuvassa 22 kuvataan DIM-osuus liitteen 1 FSP-prosessikaaviosta. RetrieveImage kutsuu C-GET DIMSE -palvelua ja palauttaa sen avulla saadun kuvan.

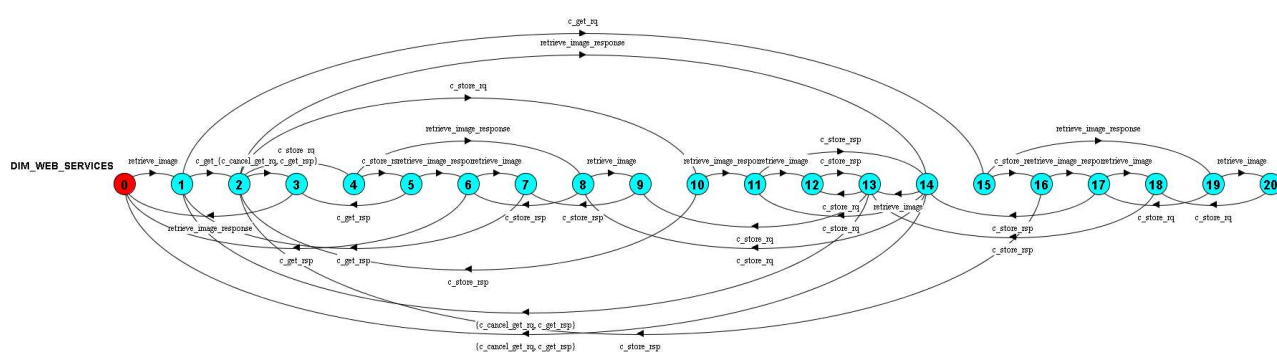
```

DIM_RETRIEVE = DIM_RETRIEVE[Request],
DIM_RETRIEVE[action:Actions] = (
  when (action==Request)
    retrieve_image -> c_get_rq -> DIM_RETRIEVE[Response]
  /when (action==Response)
    retrieve_image_response -> DIM_RETRIEVE
).

```

Kuva 22. Prosessikaavio DIM Web Service -ratkaisusta.

Kuvassa 23 on kuvattu LTS-diagrammin avulla DIM Web Service -palvelun toiminta. Kun palvelin saa RetrieveImage HTTP-pyynnön, se lähettää DICOM-verkkoon C-GET -pyynnön. Palvelin toimittaa vastauksen erillisellä RetrieveImageResponse HTTP-pyynnöllä.



Kuva 23. RetrieveImage -komento yhdistettynä C-GET DIMSE -palveluun.

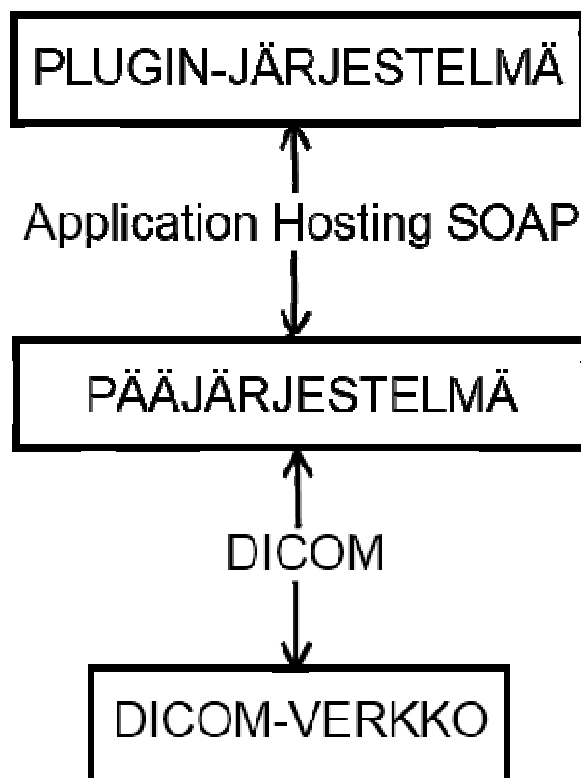
Suurin ongelma DIM Web Service -palvelussa on transaktioiden puute. Ainoa transaktio, johon otetaan kantaa, on C-Find -hakujen keskeyttäminen. C-Store -komennon samanaikaisuutta voidaan kuitenkin yrittää hallita esimerkiksi erillisten tarkistusten avulla. Tämä voidaan tehdä esimerkiksi niin, että ennen kuin dataa lähetetään, tehdään datan oikeellisuustarkistukset C-Find -palvelun avulla. Rajapinnassa voitaisiin kuitenkin ottaa käyttöön esimerkiksi käyttäjän muokkaaman tutkimusdatan lukitseminen käytön ajaksi.

Mobiiliohjelmistojen näkökulmasta DIM Web Service tarjoaa käyttökelpoisen rajapinnan esimerkiksi kuvien noutamiseen palvelimelta. Sen sijaan transaktioiden puute johtaa siihen, että kuvien tallennus palvelimelle on tämän rajapinnan kautta hieman epäsuotavaa. DIM-palvelun jatkokehityksessä transaktioiden käsittely olisi kuitenkin hyvä lisätä sen toimintaan, jolloin myös tallennusominaisuutta voitaisiin käyttää.

DIM Web Service voisi periaatteessa olla käytössä esimerkiksi WWW-verkossa. Avoimen DIM-palvelun ongelmaksi saattaa muodostua tietoturva. Tietoturvan kannalta tarpeeksi hyvään ratkaisuun voitaisiin päästä käyttäjän tunnistautumisen ja esimerkiksi potilastietojen salauksen avulla. Myös HTTPS-protokollan käyttö on DIM Web Service -palvelua käytettäessä tietoturvan kannalta suotavaa.

6.8 Application Hosting

NEMA PS 3.19 [2011] kuvaama Application Hosting on tarkoitettu kahden ohjelmiston liittämiseksi toisiinsa. Ratkaisussa toinen ohjelmisto on varsinainen pääohjelmisto ja toinen plugin-tyyppinen lisäosa. Kahdesta ohjelmistosta muodostuu tällöin käytännössä yksittäinen järjestelmä. Pääjärjestelmä tarjoaa plugin-järjestelmälle esimerkiksi kuvia ja muuta tutkimustietoa. Plugin-järjestelmä voi myös arkistoida kuvadataa pääjärjestelmään. Kuvassa 24 kuvataan, kuinka Application Hosting toimii pääjärjestelmän ja plugin-järjestelmän välisenä rajapintana.



Kuva 24. Application Hosting -rajapinta.

Plugin-järjestelmä avataan erillisessä SOAP-kutsussa pääjärjestelmän kautta. Plugin-ohjelmistolla on aina määrätty tila, jonka pääjärjestelmä voi joko hakea getState()-kutsulla tai asettaa setState()-kutsulla. Jos plugin-järjestelmässä vaihtuu tila ilman erillistä pyyntöä, tilan vaihtuminen ilmoitetaan pääjärjestelmään notifyStateChanged()-metodilla. Pääjärjestelmä voi hallita esimerkiksi plugin-järjestelmän transaktioita näiden tilatietojen avulla. Jos plugin-ohjelmasta on avattu yhteys palvelinohjelmaan, oletustila on IDLE. Muista tiloista siirrytään aina IDLE-tilaan, kun toiminto on suoritettu tai peruutettu. Muita tiloja ovat EXIT, INPROGRESS, SUSPENDED, COMPLETED TAI CANCELED. Application Hosting sisältää myös funktioita siihen, kuinka plugin-ohjelman käyttöliittymää hallitaan. Esimerkiksi pääjärjestelmä voi asettaa plugin-ohjelman käyttöliittymässä päällimmäiseksi.

Application Hosting -ratkaisun Web Service -rajapinta eroaa muiden tässä kappaleessa tarjotuista ratkaisuista. Aikaisemmat ratkaisut perustuvat avoimeen rajapintaan, jota voidaan kutsua useista järjestelmistä. Application Hosting -ratkaisussa sen sijaan on kyse kahden ohjelmiston yhdistämisestä. Esimerkiksi aikaisemmin esitelty DIM Web Service on esimerkki avoimesta

rajapinnasta. DIM Web Service -palvelussa on kuitenkin transaktioiden puuttuminen. Sen sijaan tässä Application Hosting -ratkaisussa voidaan samanaikaisuutta hallita helposti transaktiotasolla.

Tässä kappaleessa kuvattiin erilaisia XML-pohjaisia ratkaisuja DICOM point-to-point -tyyppisen tietoliikenteen ongelmien ratkaisemiseksi. Lisäksi kuvattiin, kuinka DICOM-dataa voitaisiin esittää XML-rakenteisena datana. Seuraavassa kappaleessa mietitään sitä, mitkä voisivat olla standardin tulevaisuuden haasteita. Siinä pohditaan myös tässä kappaleessa esitettyjä ratkaisuja ja sitä, kuinka ne voisivat mahdollisesti toimia tulevaisuudessa.

7. Standardin tulevaisuus

DICOM-standardi kehitettiin 1980-luvulla kahden eri järjestön toimesta. 1980-luvun jälkeen on nähty suuri kasvu tietoliikenteen määrässä ja verkkoliikennetekniikoiden kehittämisessä. Kasvu on muuttanut myös tarpeita DICOM-standardin suhteen. Tulevaisuudessa avoin tietoliikenne tulee todennäköisesti olemaan se, johon suuntaan DICOM-standardia kehitetään. Tämän takia tietyt rajoitteet DICOM-standardin rakenteessa muodostavat yhä suuremman ongelman. Ongelmana on point-to-point -tyyppinen tietoliikenne, johon DICOM-standardi nojautuu erittäin vahvasti. Varsinaisen DICOM-koodatun datan puitteissa onkin mahdotonta kuvitella laajennusta avoimempaan tietoliikenteeseen. Ratkaisuja point-to-point -ongelmaan on kuitenkin kehitetty ja kehitetään edelleen. Nykyään on olemassa useita työkaluja, joilla voi muuntaa DICOM-dataa XML-muotoon ja DICOM-pyyntöjä erilaisiksi HTTP-kutsuiksi.

Tässä kappaleessa mietitään teknisten ratkaisujen lisäksi sairaalaympäristöjen tarpeita kuvantamisdatan näkökulmasta. Kappaleessa pohditaan, kuinka DICOM-standardia voitaisiin kehittää siihen suuntaan, että se tukisi paremmin sairaaloiden prosesseja. Yksi suuri haaste tulee todennäköisesti olemaan yhä korkeampilaatuisemman kuvadatan liikuttaminen mobiiliverkossa. Tässä kappaleessa mietitään myös mobiiliratkaisuja XML-datan ja HTTP-pyyntöjen kannalta. Lopuksi kappaleessa tarkastellaan kansallisen arkiston suunnitelmia DICOM-standardin näkökulmasta.

7.1 Kuvantamisen tarpeet tulevaisuudessa

Näkyvin kehityssuunta tutkimusdatan liikuttamisen kannalta on avoimeen verkkoon siirtyminen. Tutkimusten selaaminen WWW-verkon välityksellä tarjoaa radiologille tai lääkärille mahdollisuuden, että hänen ei ole diagnoosia tehdessään pakko olla sairaalassa omalla päätteellään. Lääkäri tai radiologi voi esimerkiksi muodostaa VPN-yhteyden sairaalan sisäverkkoon ja katsella sitä kautta kuvia mobiililaitteella tai kannettavalla tietokoneella.

Avoimemman verkon ongelmana on DICOM-tyyppisen tietoliikenteen rajoittuneisuus. Tästä huolimatta sairaalan sisäinen DICOM-verkkoliikenne ja kuva-arkistointi tullaan kuitenkin todennäköisesti jatkossakin hoitamaan perinteisen DICOM UL -protokollan avulla. Tähän on erilaisia syitä, jotka liittyvät vanhoihin kuvantamislaitteisiin ja niiden laitevalmistajien ohjelmistoihin. Kuvantamislaitteet ovat usein vanhoja ja niiden päivittäminen on työlästä ja kallista. Lisäksi vanhimmissakin kuvantamislaitteissa on useimmiten tuki DICOM UL -protokollan käyttöön. Joissain tilanteissa se tosin täytyy tilata ja asentaa erikseen. DICOM-standardin taaksepäin yhteensopivuus myös varmistaa sen, että vanhojen laitteiden DICOM-tietoliikenne toimii nykyaikaisessakin DICOM-verkossa. Ainoa vaatimus on, että laitteessa on TCP/IP-protokollaa tukeva verkkokortti. Koska vanhoja kuvantamislaitteita on edelleen käytössä runsaasti, on tärkeätä, että löydetään keinot myös niillä otetun tutkimusdatan siirtämiseksi avoimempaan verkkoon. Edellisessä kappaleessa kuvatut Web Service -ratkaisut ovat varsin hyvä työkalu tällaisessa tilanteessa.

Sairaaloiden arkistointi tulee entistä enemmän painottumaan digitaalisiin kuviin fyysisen arkiston sijaan. Koska kuvat ovat jatkossa digitaalisessa muodossa, ei ole helppo löytää syitä, että miksi kuvat tällöin säilytettäisiin vain yhdessä paikassa. Tutkimusdatan olisi tärkeätä olla avointa niin, että potilaan tutkimuksia voisi katsoa myös, jos hän joutuu toisen sairaalan hoidettavaksi. Tämän takia on tärkeätä luoda tapoja, joiden avulla kuvat ovat tarjolla esimerkiksi kansallisen arkiston kautta kaikkialta maan sisältä. Voidaankin siis pitää todennäköisenä, että XML-pohjaisia tiedonsiirtoratkaisuja tullaan kehittämään jatkossa entistä paremmiksi.

7.2 DICOM ja mobiililaitteet

Radiologisten tutkimusten katseleminen ei nykyään rajoitu pelkästään sairaalan sisäiseen verkkoon. Kuvia katsellaan esimerkiksi kannettavilla tietokoneilla. Lisäksi suosiota ovat kasvattaneet mobiililaitteet. Mobiililaitteet tarjoavat tutkimustiedon selaamisen missä tahansa mobiiliverkon kantoalueella. Jo nykyisellään on olemassa erilaisia käyttökelpoisia ohjelmistoja, joiden avulla voidaan siirtää kuvia eri tavoin erilaisille mobiililaitteille kuten esimerkiksi tablet-tietokoneille. Todennäköisesti tällaiset ohjelmistot lisääntyvät ennestään, koska niille on selkeästi kysyntää.

Yksi käyttökelpoisin DICOM-mobiilisovellus nykyisellään on muun muassa Applen iPad-laitteilla toimiva OsiriX. OsiriX-ohjelmistolla voidaan katsella DICOM-kuvia joko DICOM-verkon kautta tai sen oman XML-palvelun kautta. Kyseinen sovelluksen XML-palvelu tarvitsee erillisen OsiriX-palvelimen vastapäähän. OsiriX on jo nykyisellään toimiva sovellus, mutta kuvan laatu iPad-laitteella ei vastaa sairaalan päätteiden näyttöjä. OsiriX-ohjelmistoa ei kuvan heikon laadun vuoksi voikaan käyttää diagnoosien tekemiseen. OsiriX-ohjelmiston DICOM-tuki on rajoittunut vain kuvien selaamiseen. Esimerkiksi Structured Reporting -tuki olisi hyvä lisä ohjelmistoon.

Kuten OsiriX-ohjelmistossakin, yleisimpänä haasteena mobiililaitteikäytössä voidaan pitää kuvan tasoa. Kuvanlaadun heikkouteen vaikuttaa kaksi seikkaa. Toinen on mobiililaitteiden näyttöjen laatu, joka ei vastaa harmaasävykuvien suunniteltuja sairaalapäätteitä. Lisäksi esimerkiksi häviöllisellä JPEG-kuvanpakkausalgoritmeilla kuvat muuttuvat usein rakeisiksi ja niiden harmaasävyjen määrä ei vastaa alkuperäistä kuvaa.

Yksi suuri ongelma mobiililaitteissa on langattoman verkon hitaus verrattuna fyysiseen verkkoon. Esimerkiksi sairaalan DICOM-verkon nopeus on todennäköisesti huomattavasti suurempi, kuin vaikkapa 3G-verkon tarjoama nopeus. Voidaan ajatella esimerkiksi 100 megatavun magneettikuvauksen siirtoa 3G-verkon välityksellä. Tällaisen datapaketin langattomaan siirtämiseen tarvitaan jo erittäin nopea yhteys.

Langattoman verkon nopeutta voidaan kuitenkin parantaa ottamalla käyttöön uusia tehokkaita kuvanpakkausalgoritmeja. Aikaisemmin kuvattu JPEG 2000 Streaming on yksi tärkeä ominaisuus, jonka toivoisi yleistyvän mobiililaitteiden DICOM-ohjelmistoissa. Streaming-ominaisuudella voidaan ladata niin sanottuja ROI-osia kuvasta, jolloin kuvadataa ei välttämättä tarvitse ladata kokonaisuudessaan. Tämä nopeuttaa kuvien selaamista hitaammissa verkoissa.

Kun häviöttömät kuvanpakkausalgoritmit kehittyvät, tulevaisuudessa nähdään entistä pienempään tilaan pakkautuvia kuvia. Myös pelkkiin kuvantamislaitteisiin suunniteltu

kuvanpakkausalgoritmi voisi olla tarpeen. Kuvantamislaitteissa käytetään pääasiassa pelkkiä harmaasävyjä, joten harmaasävyihin perustuva kuvanpakkausalgoritmi todennäköisesti pienentäisi kuvan vaatimaa kokoa entisestään.

7.3 DICOM-verkkojen tulevaisuus terveyslaitoksissa

DICOM-standardia käytetään Suomessa useimmissa suurissa sairaaloissa kuvien arkistointiin. Standardin avulla tutkimusdataa liikutetaan PACS-arkiston, modaliteettien ja työpisteiden välillä. Lisäksi standardia käytetään esimerkiksi HL7-standardin kanssa kuvadatan siirtämiseen. Joissain tilanteissa PACS-arkistot integroituvat esimerkiksi RIS-järjestelmän ja muiden sairaalan järjestelmien kanssa. Integraatioiden tarve kasvaa todennäköisesti tulevaisuudessa ohjelmistojen laajetessa ennestään erikoisalojen järjestelmiksi.

DICOM-standardi on alun perin suunniteltu sairaaloiden hoitoprosessien tueksi. Hoitoprosessin tuessa löytyy kuitenkin parantamisen varaa. Suurimmat ongelmat lienevät potilaan hoidon alussa ja lopussa. Hoitoprosessin alussa potilaan henkilötietojen saaminen kuvantamislaitteelle on usein vaikeata. Toisinaan radiologi joutuu kirjoittamaan tiedot käsin. Tämä saattaa johtaa virheisiin potilaan tiedoissa ja tällöin jälkepäin tutkimusta ei välttämättä pystytäkään löytämään. Tähän avuksi luotu Modality Worklist SOP -palvelun integrointi potilasjärjestelmään HL7-sanomilla tuntuu toisinaan haasteellista. Tämä saattaa kertoa siitä, että kyseinen integraatiotapa on joissain tilanteissa liian vaikea toteutettavaksi ja siihen voitaisiinkin yrittää löytää parempia ratkaisuja.

Hoitoprosessin loppupäässä olisi tärkeätä, että radiologi ja lääkäri pääsisivät mutkattomasti käsiksi kuva-arkistoon. Ongelmaksi saattaa muodostua arkiston selaaminen päätelaitteella, jota ei ole rekisteröity sallittujen yhteyksien luetteloon. Tähän avuksi voitaisiin suunnitella esimerkiksi DICOM-verkon ulkopuolinen sovellus, jota voitaisiin selata myös sairaalan ulkopuolelta esimerkiksi mobiililaitteella tai muilla langattomilla välineillä.

7.4 Kansallinen terveysarkisto

Kansallinen terveysarkisto on hanke, jota on kaavailtu Suomessa jo vuosia. Arkistossa on tarkoitus säilyttää potilasasiakirjoja, jotka tuotetaan erillisistä potilastietojärjestelmistä. Kansallisen terveysarkiston toteutumista on kuitenkin kritisoitu, koska sen toteutusaikatauluja on venytetty useaan otteeseen. Kansallisen terveysarkiston kuvadata on suunniteltu siirrettäväksi DICOM-muodossa. Määritelmän mukaan kaikki kuvadata, eli myös muu kuin modaliteeteista saadut kuvat, tallennetaan arkistoon DICOM-muodossa [Sosiaali- ja terveysministeriö, 2006]. Kuvien siirto DICOM-muodossa on perusteltua, koska kuvat säilytetään potilasasiakirjoista erillään. DICOM säilyttää kuvien tunnistetiedot ja kuva voidaan jälkepäin tunnistaa yksilöivän OID-tunnuksen avulla.

Kuvien tallennus DICOM-muodossa tarkoittaa sitä, että jos potilasasiakirjoissa on kuvia, ne on ennen lähettämistä muunnettava DICOM-muotoisiksi. Todennäköisesti niiden siirrossa käytetään SOP-luokasta hieman eroavaa DICOM-tiedostomuotoa. DICOM-tiedostot sisältävät samanlaisia VR-elementeillä koodattuja dataelementtejä, kuten palveluluokilla välitettävä data. Niiden VR-

otsikoiden tiedot kuitenkin eroavat hieman varsinaisesta DICOM UL -protokollalla välitettävästä SOP-luokasta.

Lisäksi Sosiaali- ja terveysministeriö määrittää, että jos on mahdollista, niin DICOM-muotoisesta tutkimuksesta tehdään aina erillinen CDA-dokumentti. Tämä on mahdollista luoda esimerkiksi NEMA PS 3.20 [2011] DICOM-XML -muunnoksen avulla. Sen avulla kaikki mukana oleva Structured Reportin -muotoinen potilasdata voidaan muuttaa XML-muotoiseksi, jotta se voidaan sisällyttää CDA-dokumenttiin.

7.5 DICOM-standardin kehityssuunta

DICOM-standardi on ottanut vuoden 2011 uudistuksilla harppauksen eteenpäin potilasdatan XML-muunnoksissa. Jatkossa on kuitenkin kehityttävä nopeasti, jotta saataisiin kiinni lisääntyvä tarve kuvantamisdatan avoimemmalle käytölle. Syitä tähän on esimerkiksi mobiililaitteiden nopea kehitys. DICOM-standardiin tulisi seuraavissa versioissa julkaista lisää Application Hosting -tyyppisiä erilaisia Web Service -rajapintaratkaisuja. Myös WADO-rajapintaa täytyisi joko uudistaa tai sitten se pitäisi poistaa vanhentuneena. Application Hosting -tyyppinen Web Service -rajapinta todennäköisesti olisi parempi ratkaisu kuin WADO-palvelun kehittäminen. Kyseinen Web Service -rajapinta todennäköisesti tulisi muistuttamaan myös jonkin verran tässä tutkielmassa kuvattua DIM Web Service -ratkaisua.

Siinä vaiheessa kun nykyiset kuvantamislaitteet vanhenevat ja otetaan pois käytöstä, on hyvä pysähtyä miettimään, miten dataa on järkevä liikuttaa DICOM-verkon sisällä. Jos uudet markkinoille tulevat kuvantamislaitteet siirtyvät nopeasti tukemaan Application Hosting -ratkaisua, DICOM VR -muotoinen data voitaisiin todennäköisesti vanhentaa jossain vaiheessa kokonaan. Siirtymäajalla DICOM VR -muotoisen datan taaksepäin yhteensopivuus voitaisiin tarvittaessa pitää yllä esimerkiksi tietynlaisen Broker-palvelun avulla. Kyseinen Broker vastaanottaisi vanhojen laitteiden DICOM-tulokset DICOM UL -protokollan avulla. Sen jälkeen Broker muuntaisi DICOM VR -sanomat Application Hosting -muotoiseksi, jota uudet laitteet taas tukisivat.

8. Yhteenveto

DICOM on jatkuvasti kehittyvä standardi, joka tarjoaa sekä tietorakenteen, että tietoliikennepalvelut datan siirtämiseen. Valitettavasti vanhentunut point-to-point -suunnittelumalli rajoittaa usein standardin käyttöä. DICOM perustui alun perin datan siirtoon kahden laitteen välisellä 50-napaisen kaapelin avulla. Tästä point-to-point -mallista ei ole päästy eroon, vaikka nykyään dataa siirretäänkin TCP/IP-protokollalla. Point-to-point -malli on avoimen verkon aikakaudella väistämättä liian rajoittava tapa datan siirtoon, koska se rajoittaa datan siirron kahden ohjelmiston välille.

Vaikka standardia uusitaan vuosittain, uudistukset tuovat usein vain pieniä muutoksia. Vuoden 2011 standardin uudistuksissa uudet kappaleet sen sijaan vaikuttavat olevan yleisestä verkkojen kehityksestä varsin myöhässä. Tähän ei kuitenkaan välttämättä ole syynä DICOM-standardin kehityksen hitaus, vaan verkkoliikenteen nopea ja jatkuva kehitys.

Paineet avoimempaan datan siirtoon ovat tuoneet standardiin itseensä kuvauksen DICOM-XML -muunnoksesta. Kyseinen uudistus on kovasti odotettu. Julkaisemalla tämän muunnoksen NEMA tuntuu myöntävän sen, että XML on DICOM VR -dataa selkeämpi tapa esittää sisäkkäisiin elementteihin perustuvaa tietoa.

DICOM-standardin tulevaisuuden haasteet tulevat todennäköisesti liittymään juuri erilaisiin muunnoksiin ja Web Service -palveluihin ja yleensä ottaen avoimemman tiedonsiirron tuomiin haasteisiin. Myös mobiililaitteiden tuomat haasteet vievät DICOM-standardia väistämättä kovaa vauhtia kohti XML-muotoa.

Viiteluettelo

- [NEMA, PS 3.1, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 1: Introduction and Overview, PS 3.1-2011*. NEMA, 2011.
- [NEMA, PS 3.2, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 2: Conformance, PS 3.2-2011*. NEMA, 2011.
- [NEMA, PS 3.3, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 3: Information Object Definitions, PS 3.3-2011*. NEMA, 2011.
- [NEMA, PS 3.4, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 4: Service Class Specifications, PS 3.5-2011*. NEMA, 2011.
- [NEMA, PS 3.5, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 5: Data Structure and Encoding, PS 3.5-2011*. NEMA, 2011.
- [NEMA, PS 3.6, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 6: Data Dictionary, PS 3.6-2011*. NEMA, 2011.
- [NEMA, PS 3.7, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 7: Message Exchange, PS 3.7-2011*. NEMA, 2011.
- [NEMA, PS 3.8, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 8: Network Communication Support for Message Exchange, PS 3.8-2011*. NEMA, 2011.
- [NEMA, PS 3.9, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 9: Retired, PS 3.9-2011*. NEMA, 2011.
- [NEMA, PS 3.18, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Web Access to DICOM Persistent Objects (WADO), PS 3.18-2011*. NEMA, 2011.
- [NEMA, PS 3.19, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 19: Application Hosting, PS 3.19-2011*. NEMA, 2011.

- [NEMA, PS 3.20, 2011] National Electrical Manufacturers Association (NEMA), *Digital Imaging and Communications in Medicine (DICOM), Part 20: Transformation of DICOM to and from HL7 Standards, PS 3.20-2011*. NEMA, 2011.
- [ISO, 2006] ISO 12052: *Health informatics-Digital imaging and communication in medicine (DICOM) including workflow and data management*. ISO, 2006.
- [Kaldoudi, Karaiskakis, 2006] Kaldoudi, Eleni, Karaiskakis, Dimosthenis, *A service based approach for medical image distribution in healthcare Intranets Computer Methods & Programs in Biomedicine*. February 2006, Vol. 81 Issue 2, p. 117-127, 2006.
- [Ivetic & Dragan, 2009] Dragan Ivetic & Dinu Dragan, *Medical Image on the Go! Journal of Medical Systems* Volume 35, Number 4, p. 499-516, DOI: 10.1007/s10916-009-9386-2 2009
- [Magee & Kramer, 2006] Jeff Magee & Jeff Kramer, *Concurrency: State Models & Java Programs, second edition*. Wiley, 2006.
- [Pianykh, 2008] Oleg S. Pianykh, *Digital Imaging and Communications in Medicine (DICOM)-A Practical Introduction and Survival Guide*. Springer-Verlag Berlin Heidelberg, 2008.
- [Jumppanen, Jaatinen, Toivanen 2005] Anne Jumppanen, Sami Jaatinen, Teemu Toivanen, *DICOM XML*, Seminar in Medical Informatics, Turun yliopisto 27. 5.2005.
- [Jensen, Baumgartner, 2003] Todd Jensen, Bob Baumgartner, *A flexible, multimodality structured reporting system based on medical and networking standards*. International Congress Series August 2003, Vol. 1256, p.893-899
- [Prinz, Fischer, Schuster, 2005] Michael Prinz, Georg Fischer, Ernst Schuster: *The JAVA-based DICOM query interface DicoSE*. International Journal of Medical Informatics March 2005, Vol. 74 Issue 2-4, p. 325-333
- [Nagy, 2007] Paul Nagy: *Open Source in Imaging Informatics*. J Digit Imaging. 2007 November; 20 (Suppl 1): 1.10. Published online 2007 August 3. doi: 10.1007/s10278-007-9056-1
- [Mildenberger, Echelberg, Martin, 2002] Peter Mildenberger, Marco Echelberg, Eric Martin: Introduction to the DICOM standard, *Eur Radiol* (2002) 12:920-927 DOI 10.1007/s003300101100
- [Sosiaali- ja terveystieteiden ministeriö, 2006] Sosiaali- ja terveystieteiden ministeriö: *Kansallinen terveyshanke: Opas sähköisen potilaskertomuksen rakenteesta versio 2.00*, 28.2.2006.

Prosessikaavioiden piirtämiseen käytetty ohjelmisto: Labelled Transition System Analyser Versio 3.0 / julkaisupäivä 15.4.2006. Ohjelmisto löytyy osoitteesta <http://www.doc.ic.ac.uk/ltsa/>. Url tarkistettu 20.1.2012

UML-kaavioiden piirtämiseen käytetty ohjelmisto MagicDraw versiota 17.0.1.

Tutkielmassa mainittu DICOM-XML-muuntaja JDCM löytyy osoitteesta <http://www.oocities.org/gigiobb/> versio 1.51. Url tarkistettu 3.1.2012.

Tutkielmassa mainittu OsiriXin kotisivut löytyvät osoitteesta <http://www.osirix-viewer.com/>. Url tarkistettu 22.1.2012

Liitteet

Liite 1

```
// palveluluokka lähettää pyynnön tai vastauksen pyyntöön (tai peruuttaa pyynnön ennen vastausta)
const Request = 0
const Response = 1
const Cancel = 2
range Actions=Request..Cancel

// AE-entiteetin rooli on aina SCU tai SCP
const SCU = 0
const SCP = 1
range Roles=SCU..SCP

// Tässä määritellään millainen rooli laitteilla eri palvelukutsuissa on
ROLES_SERVICES = (c_echo_rq -> NEGOTIATION[SCU]
|c_echo_rsp -> NEGOTIATION[SCP]
|c_store_rq -> NEGOTIATION[SCU]
|c_store_rsp -> NEGOTIATION[SCP]
|c_find_rq -> NEGOTIATION[SCU]
|c_cancel_find_rq -> NEGOTIATION[SCU]
|c_find_rsp -> NEGOTIATION[SCP]
|c_get_rq -> NEGOTIATION[SCU]
|c_get_rsp -> NEGOTIATION[SCP]
|c_cancel_get_rq -> NEGOTIATION[SCU]
),

/* Palvelukutsuja tehdään DICOM UL -protokollalla.

Kutsuissa kuvataan point-to-point -datan siirto kahden
AE-entiteetin välillä, jossa toinen on SCU (Service Class User) ja
toinen SCP (Service Class Provider)

Roolit esiintyvät LTSA -kaaviossa seuraavasti:
Rooli 0 = SCU ja rooli SCP = 1.
Vastakkainen rooli saadaan kaavalla FSP-kielessä näin: [(role+1)%2].

*/
```

```

// DICOM UL-neuvottelu ja yhteyden luonti
NEGOTIATION[role:Roles] = (
  associate_accept[(role+1)%2] -> TRANSFER_DATA[role]
  |associate_reject[(role+1)%2] -> ROLES_SERVICES
  |abort[(role+1)%2] -> ROLES_SERVICES
),

// DICOM UL-datan siirto
TRANSFER_DATA[role:Roles] = (
  data_tf[role] -> RELEASE[role] | abort[(role+1)%2] -> ROLES_SERVICES),

// DICOM UL-yhteyden katkaisu
RELEASE[role:Roles] = (
  release_rq[role] -> release_rp[(role+1)%2] -> ROLES_SERVICES
  |abort[(role+1)%2] -> ROLES_SERVICES
),

// Verification SOP
C_ECHO = C_ECHO[Request],
C_ECHO[action:Actions] =(
  when (action==Request)
  c_echo_rq -> C_ECHO[Response]
  |when (action==Response)
  c_echo_rsp -> C_ECHO),

// Storage SOP
C_STORE = C_STORE[Request],
C_STORE[action:Actions] =(
  when (action==Request)
  c_store_rq -> C_STORE[Response]
  |when (action==Response)
  c_store_rsp -> C_STORE),

// Query SOP (ja C-Cancel).
// Tätä käyttää myös Modality Worklist SOP
C_FIND = C_FIND[Request],
C_FIND[action:Actions] =(
  when (action==Request)

```



```

c_find_rq -> C_FIND[Response]
|when (action==Response)
c_find_rsp -> C_FIND
|when (action==Response)
c_cancel_find_rq -> C_FIND
).
```

```

// C-Get SOP (ja C-Cancel).
C_GET = C_GET[Request],
C_GET[action:Actions] =(
when (action==Request)
c_get_rq -> c_store_rq -> c_store_rsp -> C_GET[Response]
|when (action==Request)
c_get_rq -> C_GET[Response]
|when (action==Response)
c_store_rq -> c_store_rsp -> C_GET[Response]
|when (action==Response)
c_store_rq -> c_store_rsp -> c_get_rsp -> C_GET
|when (action==Response)
c_get_rsp -> C_GET
|when (action==Response)
c_cancel_get_rq -> C_GET
).
```

```

// DIM Web service, joka käyttää datan hakuun C-GETiä
DIM_RETRIEVE = DIM_RETRIEVE[Request],
DIM_RETRIEVE[action:Actions] = (
when (action==Request)
retrieve_image -> c_get_rq -> DIM_RETRIEVE[Response]
|when (action==Response)
retrieve_image_response -> DIM_RETRIEVE
).
```

```

||DICOM_SERVICES = (ROLES_SERVICES||C_ECHO||C_STORE||C_FIND||C_GET).
```

```

||DIM_WEB_SERVICES = (DIM_RETRIEVE||C_GET).
```

Composition:

DIM_WEB_SERVICES = DIM_RETRIEVE || C_GET

State Space:

$3 * 7 = 2 ** 5$

Analysing using Supertrace (Depth bound 100000 Hashtable size 8000K)...

-- Depth: 0 States: 21 Transitions: 43 Memory used: 19843K

No deadlocks/errors

Analysed using Supertrace in: 0ms

Composition:

DICOM_SERVICES = ROLES_SERVICES || C_ECHO || C_STORE || C_FIND || C_GET

State Space:

$9 * 2 * 2 * 2 * 7 = 2 ** 10$

Analysing using Supertrace (Depth bound 100000 Hashtable size 8000K)...

-- Depth: 0 States: 236 Transitions: 530 Memory used: 19936K

No deadlocks/errors

Analysed using Supertrace in: 0ms