



Juha Ahola

Mittalaitteiden integrointi Continua-standardia käyttäen

Medanets ABS -järjestelmä

Mittalaitteiden integrointi Continua-standardia käyttäen

Medanets ABS -järjestelmä

Juha Ahola
Opinnäytetyö
Syksy 2010
Tietojenkäsittely
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietojenkäsittely, Digitaalisen median ja sähköisen liiketoiminnan tietojärjestelmien suuntautumis-
vaihtoehto

Tekijä: Juha Ahola

Opinnäytetyön nimi: Mittalaitteiden integrointi Continua-standardia käyttäen – Medanets ABS
-järjestelmä

Työn ohjaajat: Leo Ilkko, Hannu Kotipalo

Työn valmistumislukukausi ja -vuosi: Syksy 2010

Sivumäärä: 36 + 3 liitesivua

Medanets Oy on erikoistunut terveydenhuollon langattomien järjestelmätuotteiden suunnitteluun. Yrityksen tuote Medanets ABS -järjestelmä kykenee vastaanottamaan mittaustuloksia langattomasti erilaisilta terveydenhuollon mittauslaitteilta. Liitettäessä eri valmistajien mittauslaitteita järjestelmään tulee jokaisen laitteen rajapinta ohjelmoida erikseen. Tämän takia kuluu paljon aikaa rajapintojen ohjelmointiin.

Työn tavoite on Continua Health Alliancen standardeja käyttäen luoda yleisluontoinen ohjelmistorajapinta Continuan sertifioiduille ja Bluetoothilla varustetuille mittalaitteille sekä mahdollistaa rajapintaspesifikaatioiden luonti eri mittauslaitetyypeille. Ohjelmistokehityksen tarve vähenee, kun valmiita ohjelmistorajapintoja voidaan käyttää eri valmistajien laitteissa.

Tietoperustana työlle käytetään ohjelmointiprojektin osa-alueita asiakasvaatimuksista ylläpitoon saakka. Tekotapana on ohjelmistokehitys ammattimaisin työkaluin toteutettuna ja hyvin dokumentoituuna alusta loppuun. Aineistona käytetään Continua Health Alliancen standardidokumentteja, Bluetooth-spesifikaatioita ja ohjelmistotuotannon työtapoja.

Saavutettuina päätuloksina ovat yleisluontoinen ohjelmistorajapinta Continua Health Alliancen sertifioidujen laitteiden ja Medanets ABS -järjestelmän välillä sekä verenpainemittarin liittymän käytännön toteutus ja testaus. Tulokset ovat hyödynnettävissä käyttäessä Continuan sertifioidua Bluetooth-verenpainemittaria ja lisättäessä järjestelmään tuki muille mittalaitetyypeille. Kehitysehdotuksena työlle on lisätä tuki Continua Health Alliancen versio 2010 suuntaviivoille, soveltaa uusia suuntaviivoja ohjelmistorajapinnan edelleen kehityksessä ja poistaa välittäjälaite, kun mahdollista. Jatkotoimenpiteenä on lisätä ohjelmistoon tuki muille mittalaitetyypeille.

Asiasanat:

Ohjelmistotuotanto, Terveystieteet, Bluetooth, Standardit, Continua Health Alliance

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems, Option of Information Systems in Digital
Media and Electronic Business

Author: Juha Ahola

Title of thesis: Integration of Measurement Devices Using Continua Standard – Medanets ABS
System

Supervisors: Leo Ilkko, Hannu Kotipalo

Term and year when the thesis was submitted: Fall 2010

Number of pages: 36 + 3

Medanets Oy specializes in designing wireless system products for health care. The company's product Medanets ABS -system is capable of receiving patient measurement results wirelessly from different medical devices. Software interfaces need to be programmed individually when connecting measuring devices from different vendors to the system. Programming all interfaces separately takes a lot of working time.

The purpose of this work is to develop a generic software interface for Continua certified medical devices using Bluetooth as physical interface. This generic software interface shall enable easy implementation of other Continua certified medical device types. The need for software engineering diminishes when ready made software interfaces can be used with different vendors' devices.

The background knowledge applied for this work is the software engineering process from software requirements to software maintenance. The working method is software development using professional tools and proper documentation. Materials used are Continua Health Alliance standard documents, Bluetooth specifications and software development processes.

The achieved result is a general software interface between Continua Health Alliance certified measuring devices and Medanets ABS -system. Interface specifications for blood pressure monitor were also created and tested. The results are exploitable when using Continua certified blood pressure monitor using Bluetooth or adding support for other measuring device types in the system. Development proposals for the work are to add support for Continua Health Alliance version 2010 guidelines, apply the guidelines in the development of software interfaces and remove the intermediary device when possible. Further steps are to expand support for different types of medical devices.

Keywords:

software engineering, health care, Bluetooth, standards, Continua Health Alliance

LYHENTEET

CASE - Computer Aided/Assisted Software/Systems Engineering

CHA - Continua Health Alliance

HDP - Health Device Profile

IEEE - Institute of Electrical and Electronics Engineers

ISO - International Organization for Standardization

MCAP - Multi-Channel Adaptation Protocol

PDA - Personal Digital Assistant

SDK - Software Development Kit

SIG - Bluetooth Special Interest Group

USB - Universal Serial Bus

SISÄLLYS

1 JOHDANTO	7
2 CONTINUA HEALTH ALLIANCE	8
3 BLUETOOTH HEALTH DEVICE PROFILE	11
4 IEEE 11073- HENKILÖKOHTAINEN TERVEYSDATA	14
5 MEDANETS ABS -JÄRJESTELMÄ	18
6 TYÖKALUT	20
6.1 Työkaluista yleensä	20
6.2 Microsoft Visual Studio 2008	20
6.3 TestLink	21
6.4 Mantis Bug Tracker	23
6.5 TortoiseSVN	24
6.6 Prosa 2004 UML Modeler	25
7 OHJELMISTOTUOTANTO	26
7.1 Ohjelmistotuotannon osa-alueet	26
7.2 Ohjelmiston vaihejakomalli	30
8 JOHTOPÄÄTÖKSET	33
8.1 Kehitystehtävän yhteenveto	33
8.2 Jatkokehitys	34
LÄHTEET	35
LIITTEET	37

1 JOHDANTO

Ohjelmistotuotannossa uusien osien lisääminen valmiiseen järjestelmään on hyvin tavallista. Kaksikolmasosaa tuotantoon käytetystä ajasta on vanhojen ohjelmistojen ylläpitoa, muuttamista sekä jatkokehitystä. Ohjelmistojen kehitys on tehtävä tulevaisuutta ajatellen, niin että uusien osien lisääminen järjestelmään on suhteellisen helppoa ja kannattavaa. Mikäli ohjelmisto on rakennettu ilman kehitysmahdollisuuksia, ainoa vaihtoehto on uuden ohjelmistokokonaisuuden suunnittelu ja toteutus.

Opinnäytetyön toimeksiantajana toimii terveydenhuollon langattomien järjestelmätuotteiden suunnitteluun erikoistunut Medanets Oy. Kempeleessä toimiva yritys hyödyntää oululaista langatonta teknologiaosaamista ja työskentelee kansainvälisesti tärkeällä alalla.

Continua Health Alliance tuo maailmalle uusia terveydenhuollon yhteenliitettävyyden toimintoja, jotka mahdollistavat entistä paremman laitteiden yhteensopivuuden. Medanets Oy haluaa käyttöönsä Continuan ratkaisut ja soveltaa niitä omaan Medanets ABS -järjestelmäänsä. Nykyiset liitännät ovat olleet hyvin valmistajakohtaisia, mikä aiheuttaa tarpeen ohjelmoida jokaiselle laitteelle oman liitännän.

Opinnäytetyön tarkoitus on Continuan standardeja käyttäen ohjelmoida integraatio Continuan sertifioimia mittauslaitteita varten Medanets ABS -järjestelmään. Käytännössä integraatio tarkoittaa ajureita ja ohjelmistorajapintaa kämmentietokoneessa, joiden avulla yhteys muodostetaan järjestelmän ja mittalaitteiden välille. Koska kaikki Continuan hyväksymät tuotteet noudattavat samaa standardien perhettä, on mahdollista luoda liitännä yhteen mittalaitteeseen ja samaa rajapintaa voi vähäisen kehityksen avulla käyttää muissakin Continuan mittauslaitteissa.

Työssä perehdytään langattoman terveydenhoidon sovelluksiin, ratkaisuihin ja käyttämiini ohjelmistotuotannon oppeihin sekä ohjelmistotyökaluihin.

2 CONTINUA HEALTH ALLIANCE

Continua Health Alliance on 2006 perustettu voittoa tavoittelematon liittouma, jonka jäseninä toimivat terveydenhoito- ja teknologiayritykset ympäri maailmaa. Liittouman tavoitteena on parantaa henkilökohtaisen terveydenhuollon tilaa. Continua pyrkii perustamaan kotiin järjestelmän yhteentoimivista terveydenhoidon ratkaisuksista, jotka edistävät itsenäisyyttä sekä antavat mahdollisuuden todelliseen terveydenhoidon hallintaan. (Pakarinen & Pärkkä 2008, 2.)

Tavoitteeseen päästäkseen Continua Health Alliancen päämäärät ovat seuraavat:

- Kehittää suunnittelun suuntaviivat, jotka mahdollistavat toimittajien rakentaa yhteentoimivia sensoreita, kotiverkkoja ja terveydenhoitopalveluja.
- Perustaa tuotesertifiointiohjelma ja antaa lupaus yhteensopivuudesta sertifioitujen laitteiden välillä.
- Yhteistyössä hallinnon säädöstelyvirastojen kanssa antaa turvallinen ja tehokas tapa hallita monipuolisia toimittajaratkaisuja. (Continua Health Alliance 2010a, hakupäivä 30.9.2010.)

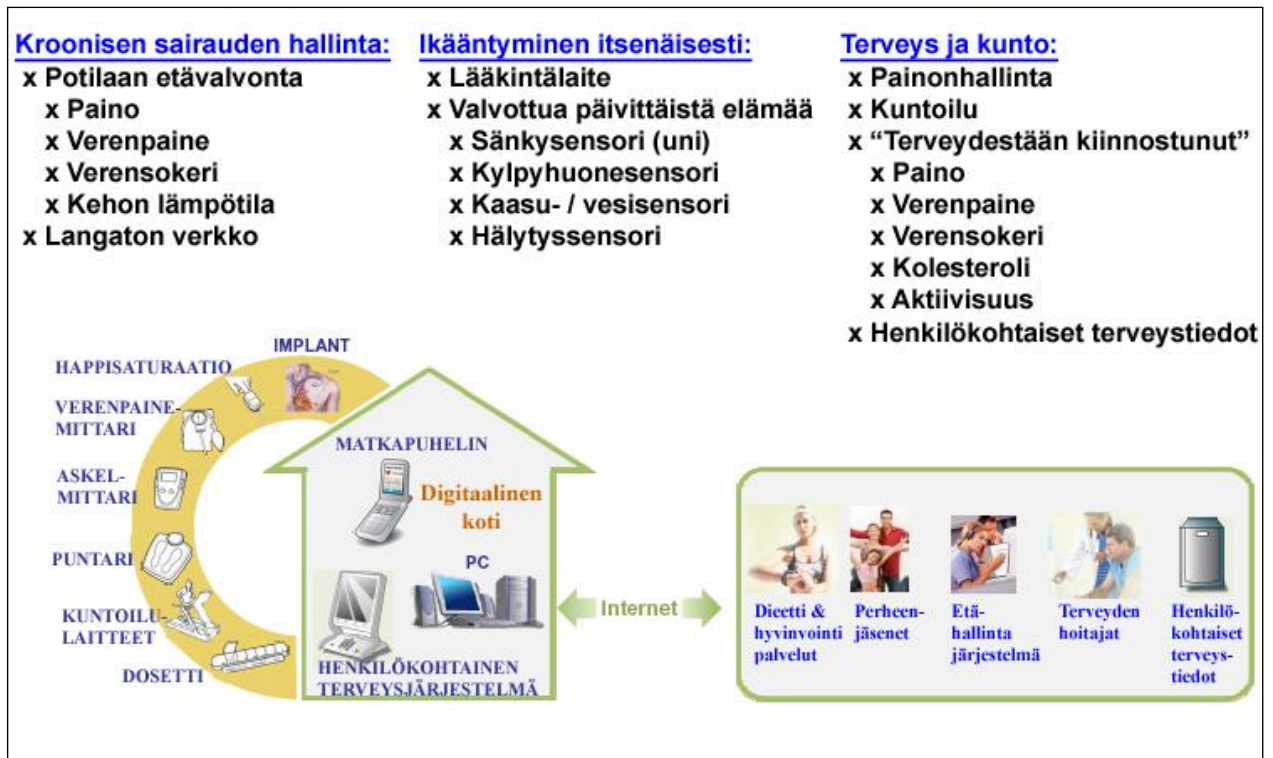
Continua-allianssi hyväksyi ensimmäisen version suunnittelun suuntaviivoista lokakuussa 2008. Suuntaviivoissa kuvataan, mitä standardeja ja spesifikaatioita käytetään varmistamaan laitteiden yhteensopivuus. Työssä syvennyttään likiverkon eli henkilökohtaisen tiedonsiirtoverkon käyttämiin standardeihin. Tiedonsiirtoon langattomasti käytetään Bluetooth Health Device Profilea ja langallisesti USB Personal Healthcare Device -luokkaa. Datatyypeihin ja muotoon käytetään Continuan kehittelemiä ISO/IEEE 11073 -standardeja. (Pakarinen & Pärkkä 2008, 9.)

Likiverkossa laitteina toimivat mittauslaitteet (Agent/Source) ja hallintalaitteet (Manager/Sink). Mittauslaite on tyypillisesti edullinen kuluttajalaite, joka omaa vähän prosessointitehoa ja muistia eikä tee muuta kuin mittauksen ja mittaustulosten edelleen välityksen. Mittauslaitteita ovat esimerkiksi verenpainemittari, vaaka ja kuumemittari. Hallintalaite omaa paljon prosessointitehoa ja muistia, koska sen pitää pystyä olemaan yhteydessä useaan erilaiseen mittauslaitteeseen, tallentaa ja esittää mittaukset käyttäjälle sekä lähettää mittaustulokset tarvittaessa eteenpäin. Tyypillisiä hallintalaitteita ovat matkapuhelin, tietokone ja kämmentietokone. (Pakarinen & Pärkkä 2008, 10; Bogia 2007, 10.)

Tuotesertifiointiohjelman kautta Continua-organisaation jäsenet hyväksyttävät omat tuotteensa osaksi Continuan sertifioituja tuotteita. Aluksi valmistaja lähettää hakemuksen sertifiointia varten oman laitteen sertifiointihallinnolle. Hakemus käsitellään ja laite siirretään Continuan testauslaboratorioon. Likiverkossa toimivista laitteista testataan, noudattavatko ne Continuan suunnittelun suuntaviivoja ja 11073-standardiryhmää sekä onko laite yhteensopiva toisen jo sertifioitun laitteen kanssa. Sertifiointiin kuuluvan tiedonsiirtoajapinnan testauksen voi hoitaa valmistaja itse tai lähettää laitteen maksulliselle testauslaboratoriolle. Testausten jälkeen sertifiointihallitus antaa päätöksen hyväksyä tai hylätä hakemus. Hylkäyksen jälkeen ristiriidat selvittämällä on mahdollista vielä sertifioida tuote. Kun valmistaja hyväksymisen jälkeen maksaa listautumishinnan, on tuote sertifioitu ja tuotteeseensa voi painaa Continuan logon. (Continua Health Alliance 2009, hakupäivä 11.10.2010.)

Vain jäsenyritykset pystyvät sertifioimaan tuotteitaan ja saavat käyttöönsä Continuan tekemät suunnittelun suuntaviivat. Jäsenet maksavat vuosittaisen jäsenmaksun riippuen jäsenyyden tyyppistä. Contributor- ja Supporting Participant -jäsenyydet maksavat yritykselle 6 500 dollaria vuosittain sekä Promoter-jäsenyys maksaa huimat 25 000 dollaria vuodessa. Contributor-jäsenet pääsevät käsiksi CHA:n ohjelmistokirjastoihin, lähdekoodiin ja testaustyökaluihin sekä voivat osallistua ilmaiseksi tai alennettuun hintaan Continuan järjestämiin tilaisuuksiin ympäri maailmaa. Promoter-jäsenet omaavat Contributor-jäsenen edut ja pystyvät äänestämään Continuan työryhmissä. Contributor-jäsenellä on myös mahdollisuus hakea työryhmien hallitukseen päättämään liittouman asioista. (Continua Health Alliance 2010b, hakupäivä 10.10.2010.)

Continua-allianssin kehittämät standardit ja sertifioimat tuotteet on tarkoitettu kotikäyttöön sekä liikkuvaan käyttöön. Allianssi on valinnut kolme käyttötapausta, joille ratkaisut on suunnattu. Niitä ovat kroonisen sairauden hallinta, terveys ja kunto sekä kotona asumisen tukeminen. Käyttötapaukset esitellään tarkemmin kuviossa 1. Kroonisen sairauden hallinnassa potilas mittaa kotonaan omaa terveydentilaansa ja mittaustulokset välitetään hoitohenkilökunnalle sairaalaan analysoitavaksi. Terveystiedot ja kunnostaan välittävä henkilö voi seurata kuntoaan kotona ja matkalla sekä automaattisesti välittää tulokset esimerkiksi henkilökohtaiselle valmentajalleen tai tietokantaan tilastoja sekä trendejä varten. Yksin kotonaan asuvan vanhuksen toimia pystyvät muut perheenjäsenet tai terveyskeskuksen henkilökunta seuraamaan erilaisin anturein ja sensorein sekä valvomaan säännöllistä lääkkeidenottoa Continuan ratkaisuilla. Mittaustulokset ja terveystiedot välitetään eteenpäin Internet-yhteyden kautta sairaalan tietokantaan tai muuhun terveystietopalveluun. (Pakarinen & Pärkkä 2008, 5-6.)



KUVIO 1. Esimerkkejä Continuan käyttötapauksista (vapaasti suomennettu) (Teknologiakatsaus Continuassa kehitettävistä standardeista, Pakarinen & Pärkkä 2003, 6)

Sairaalaympäristöissä on mahdollista käyttää Continuan tuotteita ja standardeja tietyin rajauksin. Mittauslaitteet pystyvät muodostamaan yhteyden vain yhteen hallintalaitteeseen kerralla, mutta hallintalaite pystyy olemaan yhteydessä useaan erilaiseen mittauslaitteeseen. Kun mittauslaite on "linkitetty" yhteen hallintalaitteeseen, linkin voi purkaa vain resetoimalla mittauslaitteen. Sairaalaympäristössä, jossa hoitajilla ja lääkäreillä on omat hallintalaitteet kuten kämmentietokoneet, tulee mittauslaite resetoida, että laite yhdistää oikeaan hallintalaitteeseen. Resetoiminen lisää mittauksiin käytettävää aikaa ja hankaloittaa hoitohenkilökunnan työntekoa. Sairaalaympäristöissä yhteensopivuutta kehittävä Integrating the Healthcare Enterprise (IHE) -yhteisö on vuonna 2008 sopinut yhteistyöstä Continua Health Alliancen kanssa tarjotakseen samanlaista terveydenhuoltolaitteiden yhteensopivuutta niin kotona kuin sairaalassakin. (Bogia 2007, 10; Pakarinen & Pärkkä, 19.)

3 BLUETOOTH HEALTH DEVICE PROFILE

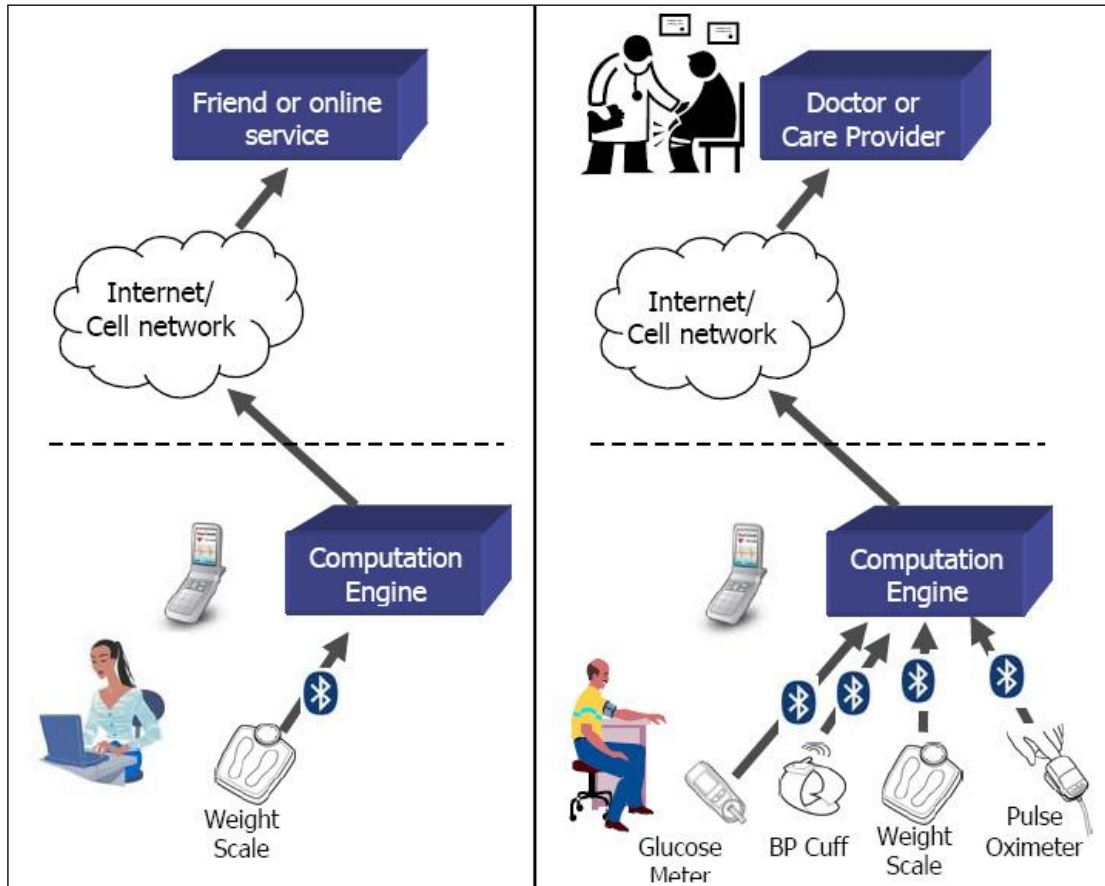
Terveydenhoitolaiteiden valmistajat käyttävät Bluetoothin langatonta teknologiaa muodostaakseen suojatun ja luotettavan yhteyden mittauslaitteen sekä mittaustenkeräyslaitteen välillä. Ennen Health Device Profilen kehitystä Bluetooth oli vain valittu ylemmän tason linkiksi laitteiden välille. Alemman tason protokollat ja formaatit olivat valmistajilleen patentoituja ja yhteistä terveydenhoitolaiteiden profiilia ei oltu määritelty. Eri valmistajien laitteet eivät olleet yhteensopivia keskenään. Laitteiden lukumäärän ja niiden luoman datan kasvaessa myös eri valmistajien laitteiden yhteensopivuus muodostui tärkeäksi. (Hughes 2009, 6.)

Yhteensopivuuden takaamiseksi Bluetooth Special Interest Group (SIG) perusti Medical Devices Working Groupin (MED WG) suunnittelemaan standardin tavan lääketieteellisen datan lähetykseen. Tuloksena syntyi Health Device Profile (HDP) ja Multi-Chanel Adaptation Protocol (MCAP). HDP toimii sidoksessa IEEE 11073 -standardiryhmän kanssa taatakseen kansainvälisen yhteensopivuuden erilaisten terveydenhoitolaiteiden kanssa. MCAP vastaa HDP:n yhteyden muodostamisesta, katkaisemisesta ja uudelleen muodostamisesta. Yhteyksiin MCAP käyttää Logical Link Control and Adaptation Protocol (L2CAP) -kanavia, joita ovat Reliable Data ja Streaming Data. (Ott 2010, 1; Hughes 2009, 10.)

Health Device Profile on suunniteltu toimimaan niin kotona, liikkeellä ollessa, klinikalla kuin sairaalassakin. Mittauslaitteena toimivat Bluetooth HDP:tä tukevat terveydenhoitolaitteet. Hallintalaitteena toimivat laitteet, jotka tukevat Bluetooth-pinoja (Bluetooth Stack) sekä vastaavat HDP:n vaatimuksiin. Työssä käytetään kämmentietokonetta hallintalaitteena ja verenpainemittaria mittauslaitteena. (Hughes 2009, 6.)

Kuviossa 2 kuvataan, kuinka erilaisiin tarpeisiin ja käyttötapauksiin soveltuu Bluetooth Health Device Profilen käyttö. Verratessa käyttötapauksia Continua Health Alliancen käyttötapauksiin nähdään, että ne ovat hyvin samanlaisia. Kuviossa 2 vasemmalla puolen terveydestään kiinnostunut henkilö käyttää puntaria. Puntarista mittaustiedot välittyvät Bluetoothia käyttäen matkapuhelimeen tallennettavaksi ja luettavaksi. Halutessaan käyttäjä voi lähettää mittaustulokset myös ystävälleen tai Internetissä toimivaan palveluun. Kuvion oikealla puolen kroonista sairautta sairastava henkilö mittaa erilaisin laittein terveydentilaansa kotona. Mittaustulokset välittyvät Bluetoothia

käyttäen matkapuhelimeen, josta mittaustulokset edelleen välitetään suojatun yhteyden yli keskitettyyn tietokantaan terveydenhoitajan tai lääkärin luettavaksi. Mittaustulosten avulla hoitohenkilökunta monitoroi potilaan terveydentilaa potilaan ollessa kotona. (sama, 7-8.)



KUVIO 2. Bluetooth Health Device Profilen käytötapauksia: Terveiden ja kunnan hallinta sekä kroonisen sairauden hallinta (HEALTH DEVICE PROFILE Implementation Guidance Whitepaper, Hughes 2009, 7-8)

Health Device Profiilin käytön etuja ovat liitettävyyys, luotettavuus, vähäinen virrankulutus ja helppo implementointi. HDP on ensisijaisesti suunniteltu ajatuksella, että kaikki profiilia käyttävät laitteet ovat yhteensopivia eikä laiteristiriitoja synny. Myöhemmin lisätyt tai kehityt laitteet eivät vaadi HDP:n päivitystä, koska uusille laitteille on valmiit liitäntätavat olemassa Health Device Profilessa. Liitettävyyys erottaa HDP:n muista Bluetooth profiileista. HDP käyttää yhteyttä muodostaessaan yhtä kanavaa kontrollikanavana ja avaa muita kanavia tarpeen mukaan datan siirtoon. Kontrollikanava toimii luotettavassa tilassa havaiten paremmin yhteyden katkomiset ja muut häiriöt, jolloin mittaustaite ja hallintalaite pystyvät paremmin reagoimaan virhetilanteisiin. Laitteet pystyvät tallentamaan nykyisen tilansa ja katkaisemaan yhteyden, kun yhteyttä ei tarvita. Kun yhteys taas

tarvitaan, edelliseen tilaan päästään nopeasti ilman ylimääräistä virran kulutusta. Yhteyden katkaiseminen ja nopea uudelleen luominen vähentää keskimääräistä virrankulutusta. HDP on optimoitu laitteille, jotka omaavat vähäiset muisti- tai prosessoriresurssit. Health Device Profile sisältää vain pienen määrän kontrollikomentoja ja tekee siitä suhteellisen edullisen ottaa käyttöön. (sama, 12-13.)

Health Device Profilea ei jokaiseen laitteeseen pystytä lisäämään, vaan HDP sisältää tiettyjä vaatimuksia laitteistolta ja ohjelmistolta. HDP vaatii toimiakseen vähintään Bluetooth version 1.2 Controllerin ja version 2.0+EDR (Enhanced Data Rate) Hostin. Käytettäväksi suositellaan Bluetooth Specification 2.1:stä, joka sisältää Secure Simple Pairing moodin yhteyden turvallisuuden lisäämiseksi. HDP ja MCAP tarvitsevat vaihtelevassa määrin laitteelta muistia, riippuen laitteen käyttötarkoituksesta. Mittauslaitteet tarvitsevat vähemmän muistia toimiakseen kuin hallintalaitteet. Laitteesta riippuen myös haihtumatonta muistia tarvitaan tietojen pitkäaikaiseen tallennukseen. Ohjelmiston toimiakseen HDP:n kanssa tulee Bluetooth pinon tukea kaikkia HDP:n ja MCAP:n pakollisia ominaisuuksia sekä L2CAP:in paranneltuja ominaisuuksia. (sama, 32-33.)

4 IEEE 11073- HENKILÖKOHTAINEN TERVEYSDATA

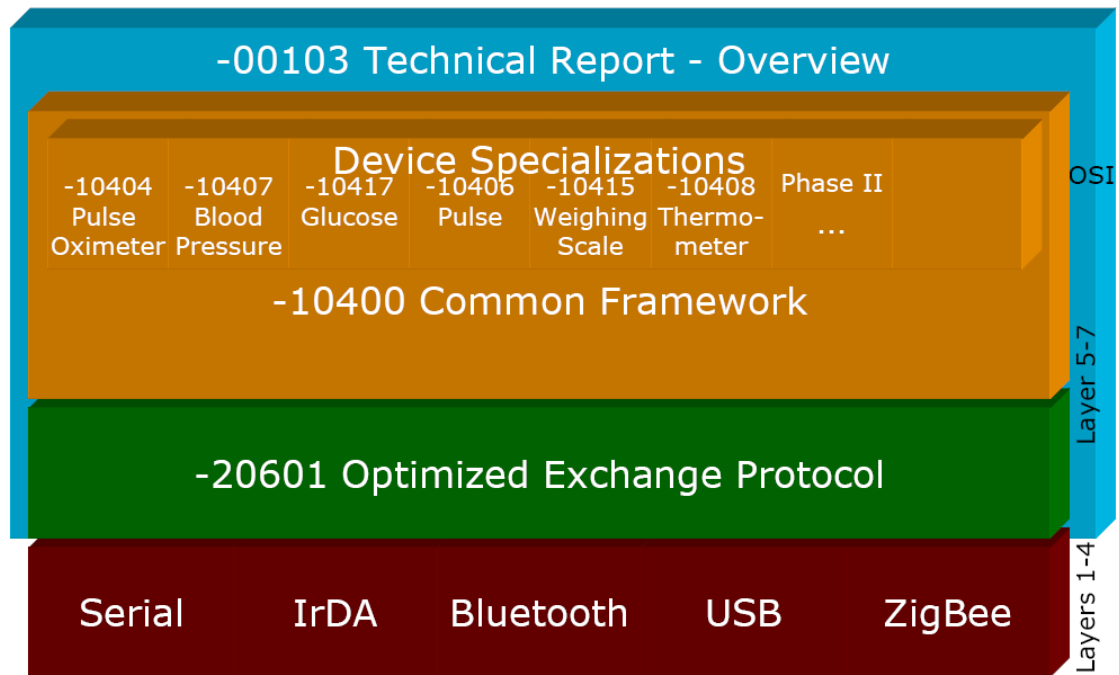
Henkilökohtainen terveyslaitetyöryhmä eli Personal Health Device Work Group toimii Engineering in Medicine and Biology Society (EMBS) alaisuudessa kehittämässä Continuan käyttämiä standardeja. Continua ei suoranaisesti vastaa 11073-standardeista vaan käyttää niitä ratkaisuisaan ja on mukana niiden kehityksessä. International Organization for Standardization (ISO) ja Institute of Electrical and Electronics Engineers (IEEE) omistavat 11073-standardiryhmän. EMSB on yksi IEEE:ssä toimivasta jaostoista. (Bogia 2007, 4.)

11073-standardiryhmä koostuu standardeista -00103 Tekninen raportti (Technical Report), -10400 Yleisrunko (Common Framework), -104xx Laitekohtaiset spesifikaatiot (Device Specializations) ja -20601 Optimoitu tiedonsiirtoprotokolla (Optimized Exchange Protocol). Tekninen raportti antaa yleiskatsauksen koko standardiryhmään ja esittelee käyttötapauksia sekä käyttömalleja. Teknisen raportin päälle rakentuvat muut standardit kuten kuviossa 3 esitetään. Yleisrunko toimii mittauslaitteiden datan pohjarakenteena, jonka päälle jokainen erityyppinen mittauslaite tuo oman spesifikaationsa. Optimoitu tiedonsiirtoprotokolla tukee useita erilaisia datatyyppejä kuten jaksottainen, tietovirta, tallennus ja edelleen lähetys. Protokolla on suunniteltu kannettavia laitteita ajatellen, jolloin datan siirron tehokkuus ja nopea uudelleen yhdistäminen ovat elintärkeitä. (Pakarinen & Pärkkä 2008, 10-11.)

Hallintalaitteelle standardit asettavat suuremmat vaatimukset kuin mittauslaitteelle. Hallintalaitteeseen täytyy pystyä yhdistämään useaan mittauslaitteeseen ja sen pitää ymmärtää eri mittauslaitetyyppien spesifikaatiot. Mittauslaitteelle riittää, että pystyy yhdistämään hallintalaitteeseen. Yhteyden vastuu on siten hallintalaitteella.

Standardit koostuvat kolmesta pääkomponentista: informaatiomallista (Domain Information Model), laitteen palvelumallista (Service Model) ja kommunikointimallista (Communication Model). Informaatiomalli kuvaa mittauslaitteen objektien ryhmänä. Jokainen objekti sisältää yhden tai useamman attribuutin. Attribuutit ovat mittauslaitteen välittämiä mittaustuloksia hallintalaitteelle ja ohjaavat sekä raportoivat mittauslaitteen tilaa hallintalaitteelle. Laitteen palvelumalli kuvaa komentotapoja, joilla mittauslaite ja hallintalaitteet keskustelevat keskenään kuten, Get- ja Set-komennot. Kommunikaatiomalli kuvaa yhden tai useamman mittauslaitteen yhteyttä yhteen hallin-

talaitteeseen. Laitteiden tilat yhteyden muodostamisesta yhteyden katkaisuun, tilojen vaihdokset ja virhetilanteet on kuvattu kommunikointimallissa. (Bogia 2007, 17, 35, 43.)



KUVIO 3. IEEE 11073- ”Henkilökohtainen terveystiedot” -työryhmän standardit (Teknologiakatsaus Continuassa kehitettävistä standardeista, Pakarinen & Pärkkä 2008, 10)

Medanets ABS -järjestelmän integraatiossa verenpainemittari on ensimmäinen laite, jonka laitespesifiset asetukset on ohjelmoitu hallintalaitteeseen. Kun verenpainemittarilla suoritetaan mittaustulos, verenpainemittari lähettää mittaus tulokset hallintalaitteelle. Mikäli yhteyttä hallintalaitteeseen ei voida muodostaa, jäävät mittaus tulokset verenpainemittarin muistiin jonoon. Kun yhteys muodostetaan, koko jono lähetetään hallintalaitteelle ja jono tyhjennetään mittauslaitteessa. Ennen ensimmäistä yhteyden muodostusta täytyy verenpainemittari ”rekisteröidä” hallintalaitteeseen. Verenpainemittari aloittaa yhteyden muodostamisen ja lähettää konfiguraatitiedot hallintalaitteeseen. Onnistuneen rekisteröinnin jälkeen verenpainemittari ottaa yhteyden aina samaan hallintalaitteeseen mittaus tuloksia lähettäessään.

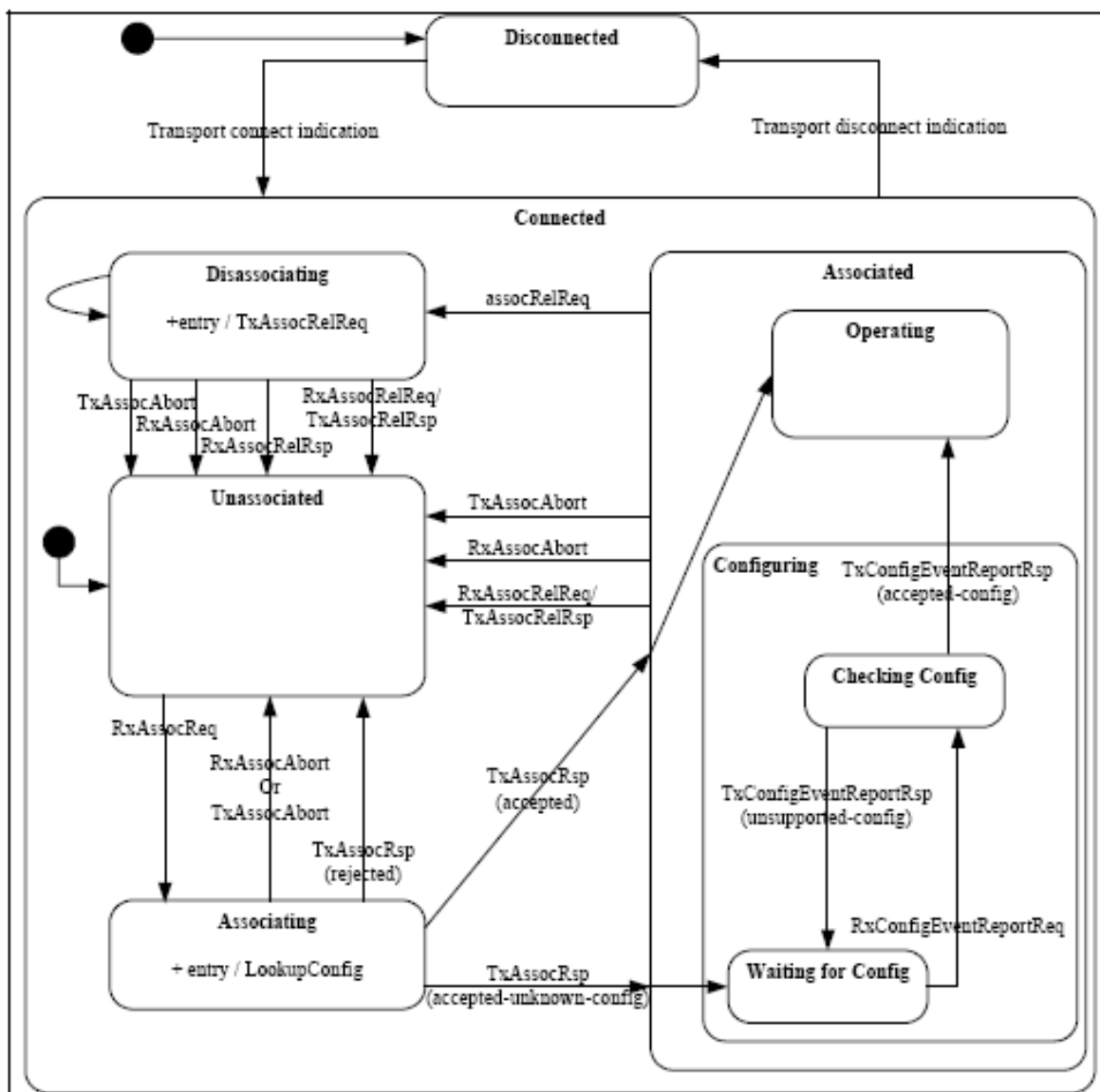
IEEE 11073 -standardiasiakirjat ovat pitkiä ja tarkkoja kuvatessaan standardien rakennetta. Koska asiakirjat kuvaavat samaa standardiperhettä, on sisällöissä paljon samaa asiaa ja useassa kohtaa on viitattu toiseen saman standardiperheen asiakirjaan tarkempien tietojen löytämiseksi.

Standardiasiakirjat sisältävät kattavat binääriesimerkit, koodaussäännöt ja nimitysten standardia käyttävän avuksi. Liitteessä 1 on esimerkki verenpainemittarin lähettämästä mittaustuloksesta.

Hallintalaitteen ja mittauslaitteen yhteydenmuodostaminen on usean eri tilan prosessi. Standardin mukainen hallintalaitteen tilakonekaavio on nähtävissä kuviossa 4. Yhteys alkaa Disconnected-tilasta, kun hallinta- ja mittauslaitteisiin kytketään virta. Kun alemman tason tiedonsiirtoprotokolla esimerkiksi Bluetooth- tai USB-yhteys on muodostettu laitteiden välille, siirrytään Connected-tilaan. Disconnect-tilaan takaisin siirtyminen tapahtuu mikäli yhteys tarkoituksella tai tahattomasti katkeaa. (Technical Committee ISO/TC 215 "Health informatics" 2009, 57, 60.)

Connect-tilan alatilana Unassociated on ensimmäinen tila johon laitteet siirtyvät, kun alemman tason tiedonsiirtoyhteys on luotu. Unassociated-tila toimii paluutilana, mikäli toinen tai molemmat laitteista keskeyttävät kommunikoinnin. Ensimmäisenä hallintalaite jää odottamaan mittauslaitteen aloittamaan yhteydenottoa. Agentti siirtyy Associating-tilaan ja lähettää yhteydenaloituspyynnön Managerille. Pyyntö sisältää tiedot mittauslaitteen käyttämisestä versioista ja protokollista. Pynnön vastaanottaessa hallintalaite siirtyy myös Associating-tilaan ja voi joko hylätä tai hyväksyä mittauslaitteen yhteyspyynnön. Managerin hylätessä yhteydenoton voi agentti edelleen lähettää muita käyttämiään versio- ja protokollatietoja. Hallintalaitteen hyväksyessä mittauslaitteen tiedot siirtyvät laitteet Associated-tilaan. (sama, 57, 58, 60.)

Mikäli hallintalaite ei tunne mittauslaitteen konfiguraatiota, tekee hallintalaite konfiguraatiosta pyynnön ja siirtyy Configuring-tilaan. Mittauslaitteen siirryttyä Configuring-tilaan ja lähetettyä konfiguraation hallintalaite tarkistaa konfiguraation sopivuuden. Epäsopivasta konfiguraatiosta manageri lähettää tiedon agentille, jolloin agentti voi lähettää toisen konfiguraation. Hyväksytyyn konfiguraation jälkeen laitteet siirtyvät Operating-tilaan ja voivat aloittaa mittaustietojen vaihdon. Hallintalaite pystyy tallentamaan mittauslaitteen konfiguraation omaan muistiinsa, jolloin saman mittauslaitteen ei tarvitse enää lähettää konfiguraatiota tulevaisuudessa. (sama, 58, 59, 60.)

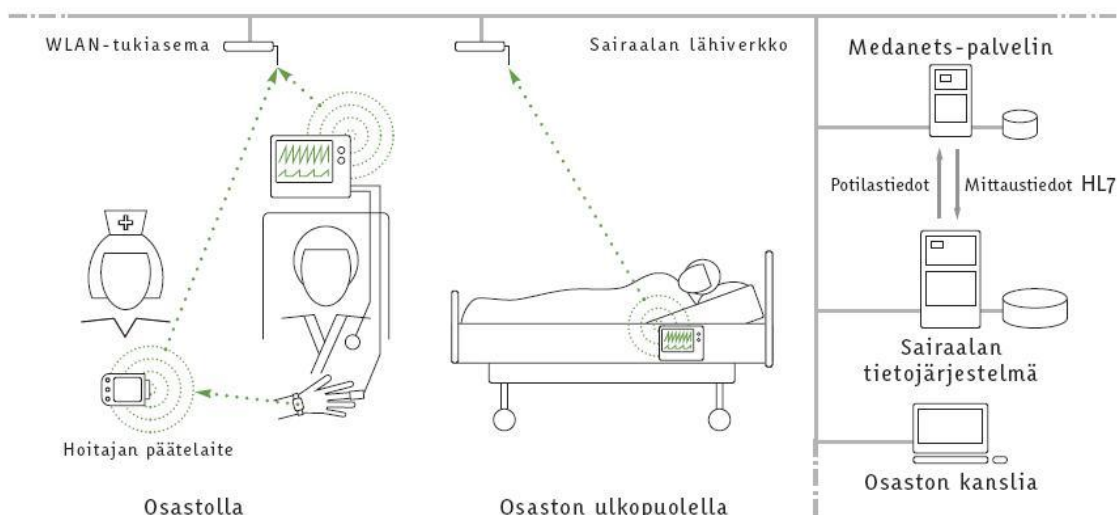


KUVIO 4. Hallintalaitteen tilakonekaavio (Health informatics - Point-of-care medical device communication - Part 20601: Application profile, Technical Committee ISO/TC 215 "Health informatics" 2009, 60)

5 MEDANETS ABS -JÄRJESTELMÄ

”Medanets Oy:n tuote on Medanets ABS - automaattisen kirjaamisen järjestelmä. Sen avulla hoitohenkilökunta voi kirjata, tallentaa ja selata potilaalle tehtyjen mittausten tuloksia” (Medanets Oy 2010a, hakupäivä 30.9.2010).

Järjestelmä pystyy käyttämään langattomia kaupallisia potilasmonitoreita sekä kannettavia pieniä mittalaitteita mittaustietojen välitykseen tietojärjestelmään. Mitatut arvot siirtyvät langattoman verkon välityksellä suoraan tietokantoihin kaikkien järjestelmän käyttäjien saataville, kuten esitetään kuviossa 5. Potilastietoja ei tarvitse kirjata useampaan paikkaan/kertaan, jolloin ajankäyttö tehostuu ja inhimillisten virheiden tekeminen vähentyy.



KUVIO 5. Medanets ABS -järjestelmän kuvaus (Medanets järjestelmän esite, Medanets Oy 2007, hakupäivä 7.10.2010)

Mittalaitteiden lisäksi Medanets ABS -järjestelmän komponentteja ovat järjestelmää ohjaava ja tietoa tallentava Medanets ABS -palvelin, potilasvalvontamonitoriin liitettävä WLAN-lähetin/vastaanotin, standardin mukaiset WLAN-tukiasemat ja hoitajan päätelaitteena käytettävä RFID-lukijalla varustettu taskutietokone (PDA) tai kannettava tietokone. Potilaan ja mittaustiedon identifointiin käytetään RFID- tai viivakooditunnisteita. (Medanets Oy 2010b, hakupäivä 30.9.2010.)

Yhä useammat sairaala- ja terveyskeskuskäyttöön soveltuvat mittalaitteet sisältävät Bluetooth langattoman liitännän. Medanets ABS -järjestelmää käyttävien hoitajien nykypäiväiset kannettavat tietokoneet ja taskutietokoneet tukevat myös Bluetoothin käyttöä. Käyttämällä Continua Health Alliancen sertifioituja tuotteita mittauslaitteet ovat yhteensopivia keskenään tehostaen hoitajien työtehtäviä. Potilaan ollessa irti johdoista myös siirtyminen sairaalassa on helpompaa. (Medanets Oy 2010c, hakupäivä 30.9.2010.)

Potilaiden tunnistaminen tapahtuu lukemalla viivakoodi tai RFID-tunniste potilaan rannekkeesta käyttämällä kämmentietokonetta, joka on esitetty kuviossa 6. RFID on standardoitu teknologia ja RFID-tunnisteet ovat helposti hankittavia kaupallisia tuotteita. Järjestelmä tunnistaa potilaan ja mittauslaitteilta vastaanotetut mittaukselliset tulokset merkitään aina oikealle potilaalle. Virhemahdollisuuksien määrä vähenee ja potilasturvallisuus paranee. (Medanets Oy 2010a, hakupäivä 30.9.2010.)



KUVIO 6. Kuva kämmentietokoneesta ja Medanets ABS -järjestelmän käyttöliittymästä (Tuotteet, Medanets Oy 2010a, hakupäivä 30.9.2010.)

Medanets ABS -järjestelmä on tarkoitettu pääasiallisesti sairaaloiden ja terveyskeskusten vuodeosastojen käyttöön. Sairaaloissa uuden teknologian käyttöönotto on aina vaativa operaatio, sillä potilaat eivät missään vaiheessa saa joutua vaaraan johtuen teknisistä virheistä tai hoitajan huonosta perehdytyksestä. Medanets Oy käyttää ammattimaisia työkaluja sekä suunnittelun ja toteutuksen tapoja tekemään potilasturvallisia että luotettavia ratkaisuja terveydenhoitoympäristöön.

6 TYÖKALUT

6.1 Työkaluista yleensä

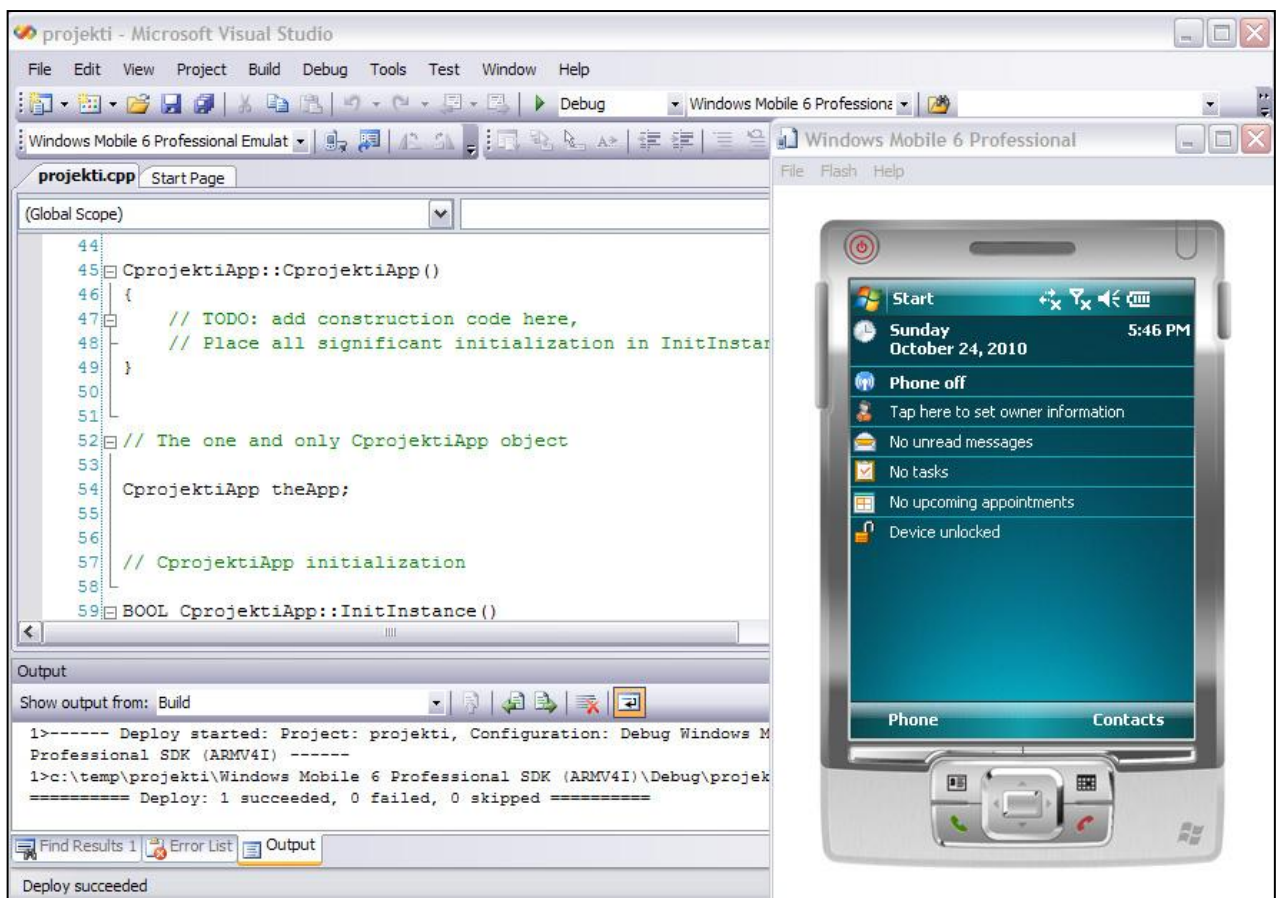
Ohjelmistotuotannon työkalut auttavat ja helpottavat ohjelmistotyön eri vaiheita. CASE-työvälineillä (Computer Aided/Assisted Software/Systems Engineering) kuvataan kaikkia ohjelmistotuotannossa käytettäviä ohjelmistoja. CASE-välineet voidaan jaotella kahteen luokkaan: edustajavälineet (upper-CASE, front-end) ja taustavälineet (lower-CASE, back-end). Edustajavälineitä käytetään ohjelmiston suunnittelu- ja määrittelyvaiheissa, esimerkiksi kun kuvataan ohjelmiston käyttötapauskäviötä. Taustavälineitä käytetään toteutusvaiheessa, esimerkiksi kun ohjelmiston ohjelmakoodi kirjoitetaan ja se käännetään kohdeympäristöä varten. (Haikala & Märijärvi 2003, 83.)

Yksinkertaisella CASE-ohjelmalla pystyy kuvaamaan ohjelmistoarkkitehtuurin eri osia, kuten arkkitehtuurisuunnittelun luokkakaavioita ja tilakonekaavioita omina kokoinaisuuksinaan. Monipuolisimmissa CASE-välineissä kerätään ohjelmistosta tehdyt kuvaukset yhteiseen kuvauskantaan, jossa voidaan tarkastaa kuvausten välistä yhteensopivuutta. Riittävän tarkoilla ohjelmiston spesifiikaatiolla voidaan monipuolisissa CASE-välineissä generoida valmista koodia ohjelmointia varten. (sama, 85.)

6.2 Microsoft Visual Studio 2008

Työssä ohjelmoinnin taustavälineenä käytetään Microsoft Visual Studio 2008 Professional Editionia. Visual Studio tarjoaa ohjelmakoodin kirjoituksen useilla ohjelmointikielillä ja kääntää usealle eri ohjelmistoalustalle. Tarjolla olevia ohjelmointikieliä ovat JScript 8.0, Visual Basic 2008, Visual C# 2008 ja Visual C++ 2008. Työssä ohjelmointi tapahtuu käyttäen C++ -ohjelmointikieltä, koska C++ tarjoaa tehokkaan ja joustavan kehitysympäristön, kun luodaan ohjelmistoja kämmentietokoneelle. (Microsoft Corporation 2008, 4.)

Visual Studio ohjelmkehitysympäristöön voidaan integroida lisäosia, kuten tässä työssä käytetty Windows Mobile 6 Software Development Kit. SDK:n avulla pystytään Visual Studiolla kehittämään ohjelmistoja, jotka toimivat Windows Mobile 6 -käyttöjärjestelmää käyttävissä kämmen-tietokoneissa. Software Development Kit antaa ohjelmistokehittäjälle käyttöön ylimääräisiä emulaattoreita ja työkaluja, kuten esitetään kuviossa 7, jotka hyödyntävät Windows Mobile 6:n ominaisuuksia. Emulaattori jäljittelee kämmentietokoneen toimintaa ja kämmentietokoneelle suunnat- tuja ohjelmia pystyy testaamaan tietokoneella ennen ohjelmiston siirtämistä kämmentietoko- neelle. (sama, 20.)



KUVIO 7. Microsoft Visual Studio 2008 Professional Edition ja Windows Mobile 6 Professional emulaattori

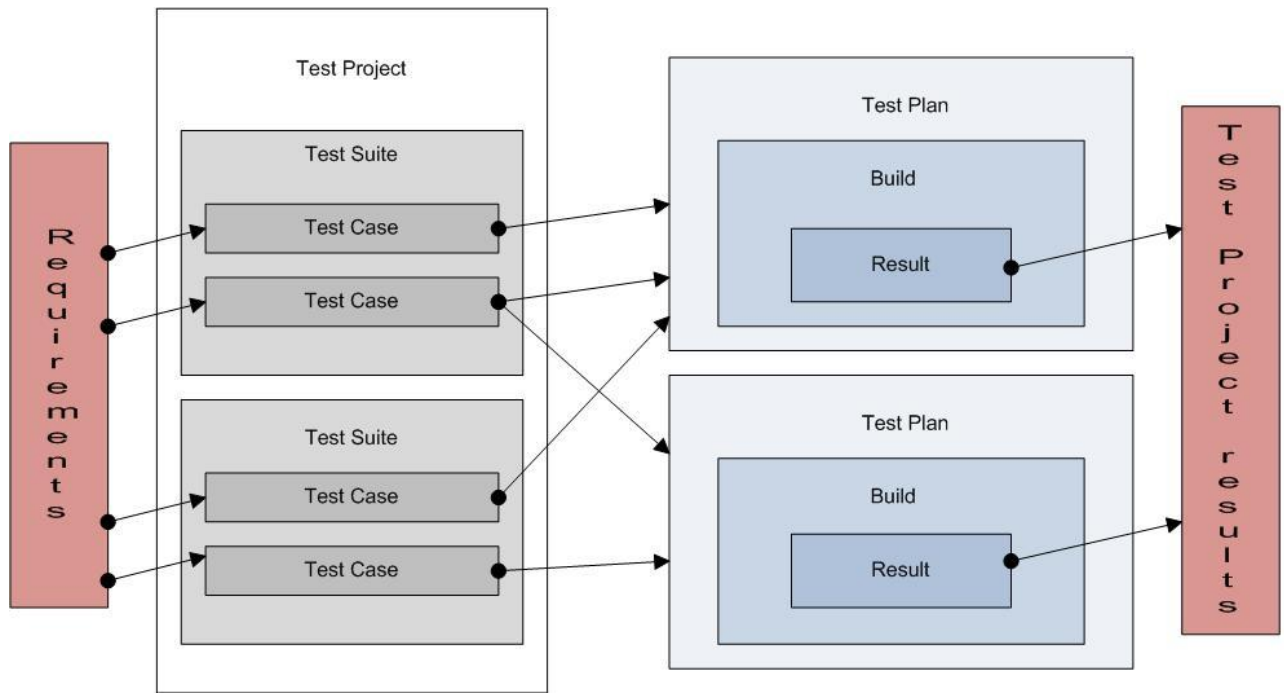
6.3 TestLink

Työn testauksen hallintaan käytetään web-pohjaista TestLink-ohjelmistoa, kuva käyttöliittymästä liitteessä 2. Web-pohjaisuuden takia TestLinkiä pystytään käyttämään millä tahansa tietokoneella, jossa on web-selain sekä yhteys palvelimeen, missä TestLink-ohjelmisto pyörii. TestLink on il-

mainen ja opensource-pohjainen eli käyttäjä voi tutustua ohjelman lähdekoodiin ja muokata sitä omien tarpeidensa mukaan. TestLink tukee myös usean käyttäjän testauksen hallintaa, jolloin kaikille ohjelmistoprojektiin kuuluville annetaan omat roolit ja määrätään työtehtävät. Roolien mukaan projektiin osallisilla on erilaisia oikeuksia ja näkökulmia projektiin. Admin omaa kaikki oikeudet testaukseen, Test Leader rakentaa testausprojektin, Test Analyst suorittaa testit sekä kirjoittaa kommentit ja Guest pääsee vain näkemään testien tulokset. Tätä työtä varten luotiin uusi testausprojekti, jossa oli kaksi Adminia ja yksi Test Leader. (Havlát, Morsing, Mancardi & Pollans 2008, 6.)

Testausprojekti sisältää yhden tai useamman Test Suiten, Test Casen, Test Planin ja Buildin, kuten on kuvattu kuviossa 8. Test Suite sisältää Test Caset ja jaottelee Caset omiin luokkiinsa esimerkiksi testattavan osion mukaan. Test Case on yksi testaus ja sisältää tiedon mitä testataan, missä ympäristössä sekä mitkä ovat odotettavissa olevat tulokset. Mikäli tulokset eivät ole mitä odotettiin, on aina kyseessä oleva Case epäonnistunut. Onnistumisen lisäksi Test Case voi olla myös estetty eli testiä ei pystytä suorittamaan ohjeiden mukaisesti. Test Planiin lisätään Test Caset, jotka halutaan sisällyttää testaukseen. Test Plan voi sisältää Test Caseja myös useasta projektista, mahdollistaen toisiinsa liittyvien projektien yhtäaikaisen testauksen. Test Plan sisältää Buildit. Build mahdollistaa testaajan kerran testata Test Planin kaikki Test Caset. Kun Build on suoritettu loppuun, täytyy tehdä toinen Build, että samat Test Caset voidaan suorittaa uudestaan. (sama, 4, 5, 14, 18.)

Työssä käytetään vaatimuksiin perustuvaa testausta ja bugitestausta. Vaatimuksiin perustuvassa testauksessa projektin jokaisesta asiakasvaatimuksista luodaan yksi tai useampi Test Case. Toivuttamalla Test Caseja vaatimusmäärittelyn mukaan nähdään, että ohjelmisto on rakennettu vaatimusten perusteella. Samalla testaus myös paljastaa epämääräisiä ja puutteellisia vaatimuksia. Bugitestausta suoritetaan, kun havaitaan ohjelmiston sisältävän semanttisen virheen. Semanttinen virhe aiheuttaa ohjelman toimimaan väärin. Bugi kirjataan Bug Tracking -ohjelmistoon ja siitä tehdään Test Case -testausprojektiin. Kun bugi on korjattu, testataan että virhe on todella kadonnut ohjelmistosta. (Havlát ym. 2008, 11; Haikala & Märijärvi 2003, 285, 288.)





KUVIO 8. Malli ohjelmiston testauksesta käyttämällä TestLink-ohjelmistoa

6.4 Mantis Bug Tracker


Tämän ohjelmiston testauksessa löydetty virheet kirjataan Mantis Bug Tracker -ohjelmistoon. TestLinkin tapaan Mantis on web-pohjainen, opensource ja pyörii palvelimella, johon otetaan yhteys web-selaimella. Mantis käyttää ohjelmointikielensä PHP: Hypertext Preprocessorin ja tukee useimpia tietokantoja sekä käyttöjärjestelmiä. (The MantisBT Team 2010, 7.)

Bugia kirjattaessa Mantis-ohjelmistoon tulee selvittää, minkälaisesta virheestä on kyse. Esimerkki kirjauksesta on kuviossa 9. Käsiteltäviä kohteita kirjauksessa ovat seuraavat: onko virhe toistettavissa, kuinka vakava virhe on kyseessä ja millä tärkeysasteella on virheen korjaus. Toisinaan bugi voi esiintyä vain silloin tällöin, kun virheen luonnetta ei vielä tiedetä, tai ohjelmointivirhe ei aiheuta ongelmaa vaan lisää ominaisuuden ohjelmistoon. Muita kirjattavia tietoja on missä virhe esiintyy, millaisessa käyttöympäristössä, mahdollisia syitä virheen esiintymiseen, virheen löytäjä ja kenelle virheen korjaus on suunnattu. Kirjauksen jälkeen bugitieto on muiden Bug Tracker -ohjelmistoa käyttävien luettavissa ja muokattavissa. Kun virhe on korjattu, tehdään alkuperäiseen bugikirjaukseen lisäys, että miten virhe on korjattu, mistä virhe johtui ja kuka korjauksen teki.



Logged in as:  (Juha Ahola - developer) 2010-10-25 18:40 EEST Project: CA Driver

[Main](#) | [My View](#) | [View Issues](#) | [Report Issue](#) | [Change Log](#) | [Roadmap](#) | [Docs](#) | [My Account](#) |
[Logout](#)

Recently Visited: 

Enter Report Details [[Advanced Report](#)]

*Category	<input type="text" value="(select)"/>
Reproducibility	<input type="text" value="always"/>
Severity	<input type="text" value="minor"/>
Priority	<input type="text" value="normal"/>
*Summary	<input type="text" value="Bug found while testing interoperability"/>
*Description	<input type="text" value="Personal Digital Assistant does not recognize other types of measurement devices."/>

KUVIO 9. Bugin kirjaus Mantis Bug Tracker -ohjelmistoon (osa kuviosta peitetty)

6.5 TortoiseSVN

Työssä versionhallinnan ohjelmistona käytetään avoimeen lähdekoodiin perustuvaa TortoiseSVN-asiakasohjelmistoa. TortoiseSVN integroituu Windowsin resurssinhallintaan käytettäväksi eli käyttäjän ei tarvitse vaihtaa muita käyttämiään ohjelmistoja versionhallinnan takia. Ohjelmisto tallentaa ja pitää kirjaa versionhallintaan lisätyistä, muokatuista ja poistetuista tiedostoista sekä henkilöistä kuka muutokset on tehnyt. Kaikki tiedostot tallennetaan keskitettyyn arkistoon, josta muut projektiin osalliset pystyvät tiedostoja hakemaan ja lisäämään sekä muokkaamaan tiedostoja. Arkistoa ylläpitää palvelin, johon asiakasohjelmat ottavat yhteyden. Kaikki versiot ja muutokset tallentuvat arkistoon ja tarvittaessa vanhempia versioita pystytään tarkastelemaan. (Küng, Onken, Large 2010, 1.)

Usean käyttäjän muokatessa samoja tiedostoja muodostuu vaaraksi, että tallentaessa muutoksia poistetaan epähuomiossa toisen käyttäjän muutokset. TortoiseSVN tarjoaa tähän ongelmaan ratkaisun tiedoston lukituksella tai yhdistämällä tiedostot. Tiedoston lukituksessa käyttäjä, joka aikoo

tehdä muutoksia, lukitsee tiedoston estäen muiden käyttäjien pääsyn tiedostoon. Tallennuksen jälkeen lukitus puretaan ja muut käyttäjät voivat lukea tapahtuneet muutokset ja tehdä omia muutoksiaan. Yhdistä tiedostot -ratkaisulla käyttäjät päivittävät arkistossa olevan tiedoston omalla versiollaan. Käyttäjän suorittaessa päivitystä verrataan käyttäjän itselleen lataamia tiedostoja arkiston tiedostoihin. Mikäli käyttäjän tiedostot ovat vanhentuneet, saa käyttäjä tästä tiedon ja mahdollisuuden verrata omaa versiotaan arkiston uuteen versioon. (sama, 5-9.)

TortoiseSVN mahdollistaa usean eri projektin arkistoinnin samaan aikaan. Projektit voivat sijaita samassa arkistossa tai omissa arkistoissaan riippuen liittyvätkö projektit toisiinsa. Arkistojen sisältö voidaan jakaa haluamallaan tavalla, mutta suositeltuna tapana on jakaa arkisto päälinjaan, kehityshaaroihin ja versioihin. Tässä työssä projektit liittyvät toisiinsa ja ovat samassa arkistossa sekä jakavat saman kehityshaaran muiden Medanets ABS -järjestelmän projektien kanssa. (sama, 17.)

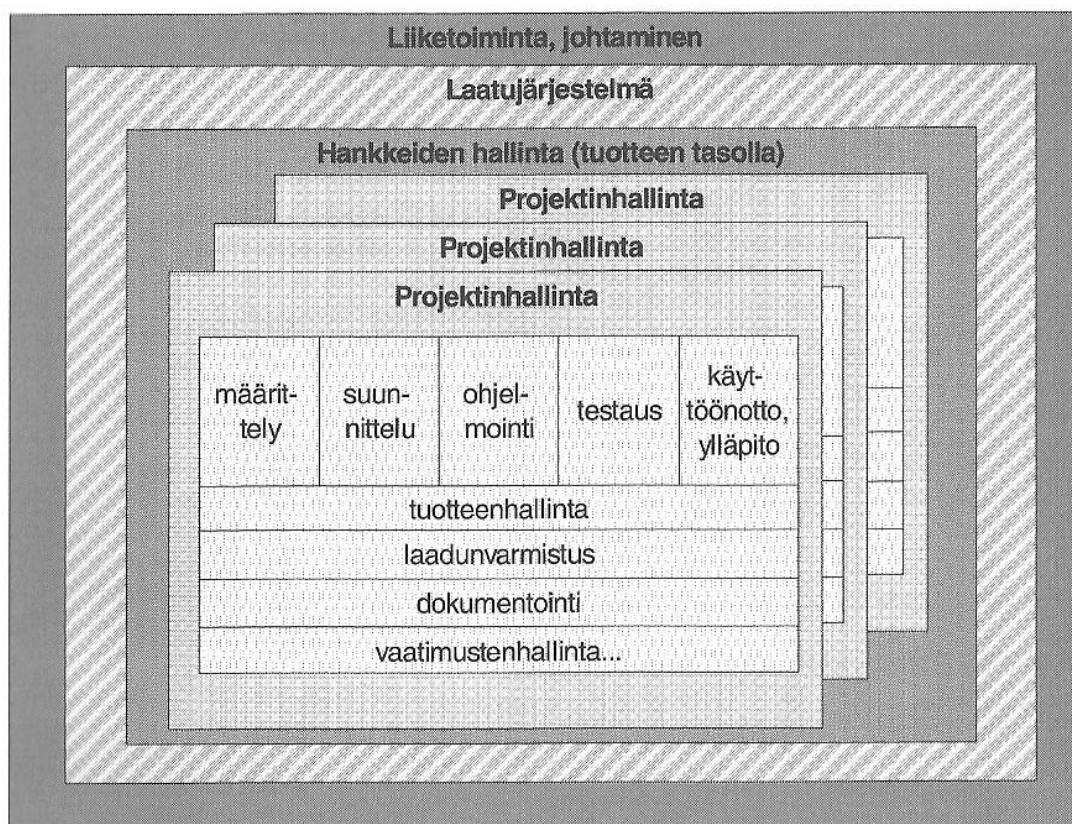
6.6 Prosa 2004 UML Modeler

CASE-välineistä edustajavälineenä työssä toimii Prosa 2004 UML Modeler. Suunnittelutyössä hahmoteltaessa ohjelmiston rakennetta ja moduuleja Prosa tarjoaa erityyppisiä kuvaamiskaavioita moduulien kuvaukseen. Valittavana on käyttötapauskaavioita, sekvenssikaavioita, luokkakaavioita, tilakonekaavioita ja muita kaavioita. Eri kaavioiden piirrostyyppejä voi yhdistellä toisiinsa luoden samaan malliin useamman kaavion. Kaaviot tukevat kuvien, videoiden ja tekstiviitteiden lisäystä. Työssä suurin osa suunnittelun kaavioista on tehty käyttäen Prosa-ohjelmistoa. Yksinkertainen sekvenssikaavio on nähtävissä liitteessä 3.

7 OHJELMISTOTUOTANTO

7.1 Ohjelmistotuotannon osa-alueet

Ohjelmistotuotanto koostuu useista erilaisista osakokonaisuuksista. Jokainen osa-alue on tärkeä koko ohjelmistoprosessin kannalta ja jokaiselle osa-alueelle on usein oma työkalunsa käytettäväksi. Tuotanto voidaan jakaa kuviossa 10 esitetyllä tavalla.



KUVIO 10. Ohjelmistotuotannon osa-alueet (Ohjelmistotuotanto, Haikala & Märijärvi 2003, 35)

Tässä työssä keskityn projektinhallinnan sisältöön ja kuinka sen eri osa-alueet toteutetaan, että saadaan projekti suoritettua onnistuneesti. Projektinhallinta koostuu vaiheista määrittely, suunnittelu, ohjelmointi ja testaus, joita seuraa käyttöönotto sekä ylläpito. Tukitoimintoina projektilla on tuotteenhallinta, laadunvarmistus, dokumentointi ja vaatimustenhallinta. (Haikala & Märijärvi 2003, 35-36.)

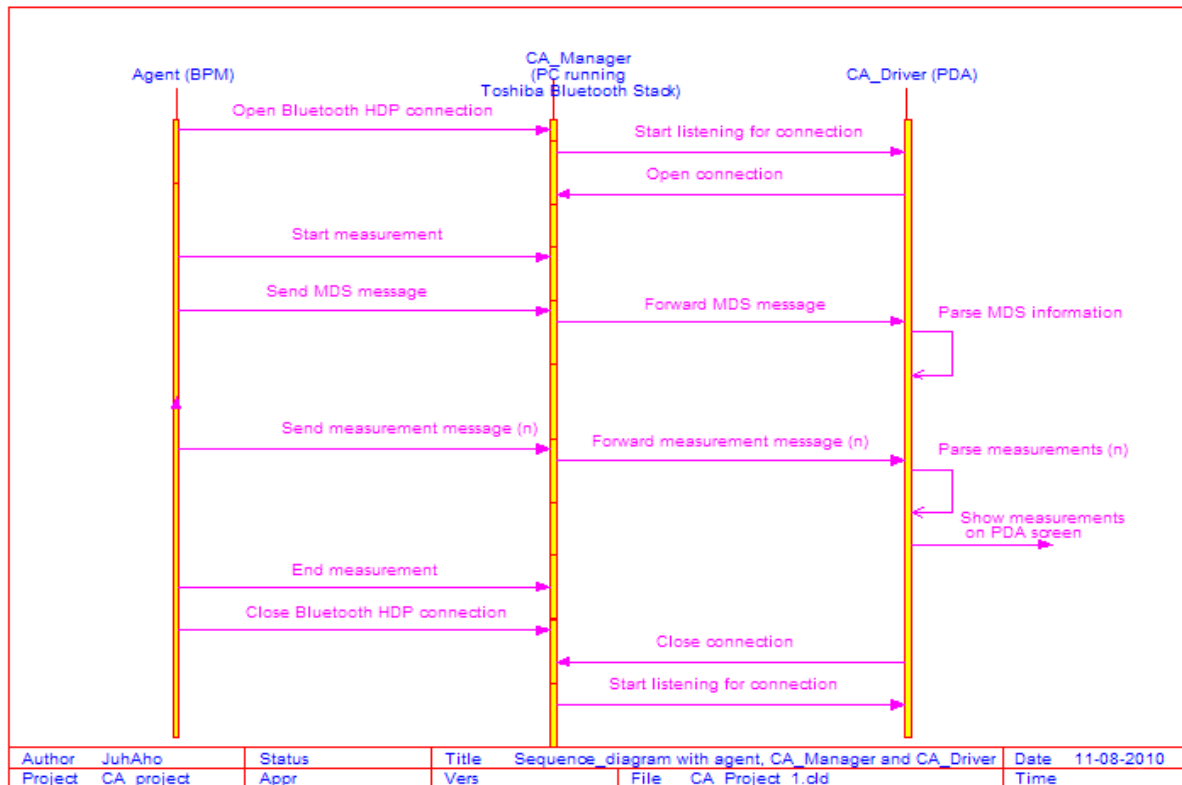
Määrittelyvaiheessa käsitellään asiakasvaatimukset, joista luodaan ohjelmistovaatimukset, jotka puolestaan määrittelevät ohjelmiston rakenteen. Asiakasvaatimukset ovat tässä tapauksessa Medanets Oy:n antamat vaatimukset, miten integraation tulee toimia. Standardien toiminnallisuuden pohjalta muokkasin asiakasvaatimuksista ohjelmistovaatimukset integraatiolle. Työkaluna työssä toimivat dokumenttipohjat ja tekstinkäsittelyohjelmat. (sama, 38.)

Suunnitteluvaiheessa määrittelyvaiheen perusteella suunnitellaan toimintojen toteutus. Aluksi toiminnot jaetaan toisistaan riippumattomiin osiin eli moduuleihin. Jakoa kutsutaan arkkitehtuurisuunnitteluksi. Seuraavaksi suunnitellaan moduulien rakenne ja sisältö. Moduulille on erilaisia tulkintatapoja, mutta tässä tapauksessa moduuliksi kutsutaan ohjelmasta erotettavaa loogista kokonaisuutta. Moduuli sisältää tietomäärittelyitä ja tietoa käsitteleviä funktioita. Työssä kuvaan ohjelmiston toimintoja käyttötapauskaavioin, sekvenssikaavioin, luokkakaavioin sekä staattisten että dynaamisten mallien avulla. Työkaluna toimii Prosa 2004 UML Modeller. (sama, 39-40.)

Ohjelmointivaiheessa työstetään ohjelmakoodia suunnitteluvaiheen moduulien mukaan. Moduulit ja niiden rajapinnat toisiinsa ohjelmoidaan ja käännetään käyttöympäristöön sopiviksi. Usein ohjelmointivaihe ja testauksen aiheuttama lisäohjelmointi ovat eniten aikaa vieviä osia ohjelmistotuotannossa. Ohjelmakoodin määrä vaihtelee moduulien koon ja monimutkaisuuden myötä. Medanets-integraatiossa ohjelmointi tapahtui käyttämällä Microsoft Visual Studio 2008 Professional Editionia. Integraation ohjelmoiminen oli aluksi hankalaa, koska kämmentietokoneen Bluetooth stack ei tukenut HDP:tä. Avuksi ongelmaan löytyi Health Device Profile -tuen sisältävä Toshiba Bluetooth stack ja Toshiba Software Development Kit, joilla HDP-liitettävyys pystyttiin ohjelmoimaan. Huonona puolena ratkaisussa olivat Toshiba SDK-funktiot, jotka asettivat rajoitteita HDP:n käyttöön sekä se, että Toshiba Bluetooth stack toimii ainoastaan Windows-PC:ssä. En pystynyt ohjelmoimaan niin vapaasti, kuin alun perin olin suunnitellut, koska sekä mittauslaitteen ja hallintalaitteen väliin tarvittiin välittäjälaite, joka ymmärsi niin mittauslaitetta kuin hallintalaitettakin. Välittäjälaitteena toimii kannettava tietokone, jossa Toshiba Bluetooth stack pyörii.

Kuviossa 11 esitetään sekvenssikaavion muodossa, miten mittauslaitteen, välittäjälaitteen ja hallintalaitteen väliset rajapinnat toimivat. Kuviossa on kuvattu yhden mittaus tiedon siirto, ja kirjain n kuvaa viestien määrää. Ensimmäisenä mittauslaite avaa HDP-yhteyden välittäjälaitteeseen ja välittäjälaite odottaa hallintalaitteen muodostavan yhteyden. Kun hallintalaitte on muodostanut yhteyden välittäjälaitteeseen, voi mittaus alkaa. Agentti lähettää tiedot itsestään ja mittaus tiedot kannettavalle tietokoneelle, joka välittää tiedot eteenpäin managerille. Manageri purkaa viestit ja

esittää tulokset näytöllään. Mittauksen loputtua mittauslaite katkaisee HDP-yhteyden ja hallintalaitte yhteyden välittälaitteeseen.



KUVIO 11. Sekvenssikaavio mittaus-, välitys- ja hallintalaitteen välisestä kommunikaatiosta

Testausvaiheessa etsitään ohjelmistovaiheessa ohjelmakoodiin eksyneitä virheitä sekä heikon vaatimusmäärittelyn aiheuttamia puutteita.

Testaus yleensä tapahtuu usealla eri tasolla V-mallin mukaisesti. V-mallissa testaus on jaettu moduulitestaukseen, integrointitestaukseen ja järjestelmätestaukseen. Moduulitestauksessa etsitään vikoja yksittäisistä moduuleista, integrointitestauksessa moduulien yhteistoiminnasta ja järjestelmätestauksessa koko järjestelmän toiminnoista ja suorituskyvystä (Haikala & Märijärvi 2003, 40.)

Testaukseen saa helposti paljon aikaa kulutettua ilman kunnon suunnitelmaa. Useita päiviä kestänyt umpimähkään kokeilu voi olla yhtä tehokasta kuin muutaman tunnin systemaattinen testaus suunnitelman mukaan. Käytin TestLink-ohjelmistoa testaukseen ja testauksen suunniteluun Medanets Oy:n kehitys- ja tutkimusprosessin mukaisesti. Käyttöön otetun Toshiba Bluetooth stackin aiheuttamat rajoitukset toivat haasteita testaukseen. Yksittäisten moduulien testaus oli hankalaa, koska SDK-funktiot tekivät itsenäisesti useat prosessit antamatta käyttäjälle tietoja prosessin ete-

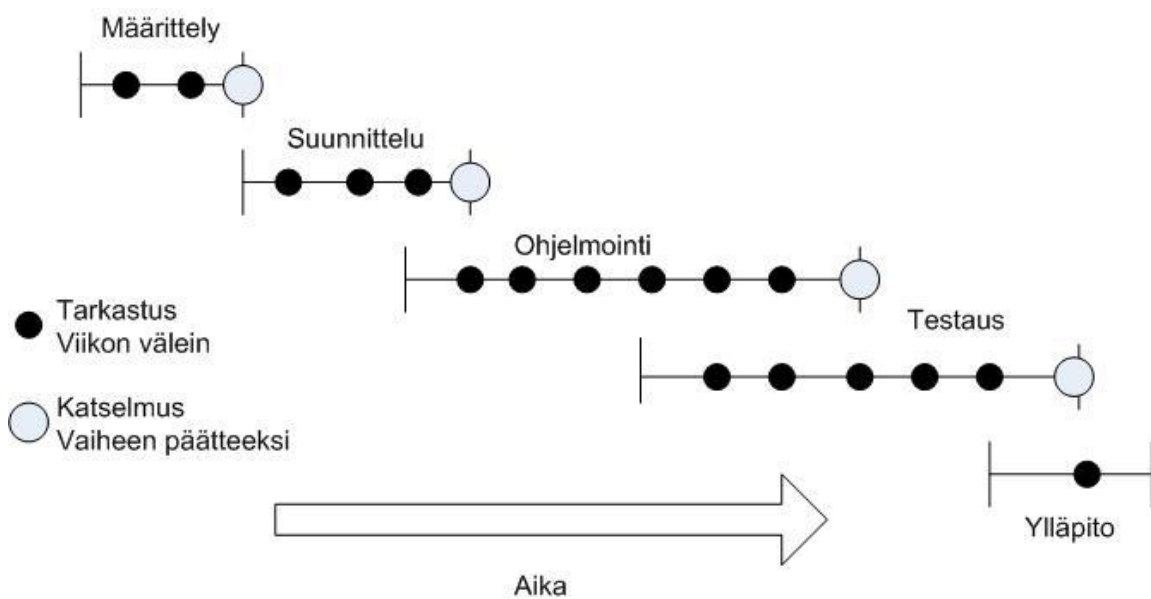
nemisestä. Esimerkiksi koko yhteydenmuodostus oli omana funktionaan eikä sen sisältämiä moduleita pystynyt testaamaan itsenäisesti. (sama, 40.)

Käyttöönotto- ja ylläpitovaiheessa valmis ohjelmisto lisätään osaksi asiakkaan jo olemassa olevaa järjestelmää ja tarpeen mukaan ratkotaan asiakkaan ongelmia, korjataan virheitä, muutetaan ohjelmaa vaatimusten muuttuessa sekä lisätään uusia piirteitä. Usein ohjelmistotuotteilla ei ole ylläpitoa ellei kehittäjä ole asiakkaan palveluksessa, muulloin ylläpito jää asiakkaan järjestelmän ylläpitäjän tehtäviksi. Korjaukset, muutokset ja lisäykset toteutetaan seuraavaan versioon ohjelmistosta, joka tarjotaan tai myydään asiakkaalle päivityksenä tai lisäosana. Useimmiten asiakas lataa ohjelmiston päivityksen itse ohjelman valmistajan WWW-sivuilta. Käytin Mantis Bug Tracker -ohjelmistoa työn ylläpidon työkaluna kirjatessani ohjelmistosta tai järjestelmästä löytämiäni virheitä. (sama, 40-41.)

Tuotteenhallinta on tukitoiminto, joka määrittelee projektin komponenttien ja konfiguraatioiden versionhallinnan menetelmät, muutosten hallinnan ja arkistoinnin. Komponentit voivat olla ohjelmistokoodia, automaattisesti tuotettavissa olevia ohjaustiedostoja ja ohjelman tekemiseen, testaamiseen sekä käyttämiseen liittyviä dokumentteja. Konfiguraatio on komponenttien kokoelma, josta tuotteen tietty versio koostuu. Tuotteenhallinnan tulee sisältää ohjelmistoprojektin kaikki komponentit ja konfiguraatiot, että tarvittaessa voidaan palata aikaisempaan versioon tai selvittää mihin ohjelmiston osiin tehtävät muutokset vaikuttavat. Työssä käytin TortoiseSVN-ohjelmistoa versionhallintaan. (sama, 51.)

Laadunvarmistus tukitoiminto toimii osana laatujärjestelmää. Laadunvarmistus keskittyy projektin edetessä syntyvien vaihetuotteiden laadunvarmistukseen. Laadua voidaan varmistaa erilaisilla laatujärjestelmän tavoilla, kuten auditoimalla, tarkastuksilla tai testauksilla. Auditoinnissa laatujärjestelmää tai jotain sen osaa käydään systemaattisesti läpi ja varmistetaan, että toiminta on laatujärjestelmän mukaista. Tarkastuksissa projektin eri vaiheisiin asetetaan katselmus- ja tarkastustilaisuuksia. Tilaisuudet on kuvattu kuviossa 12. Käytännössä tarkastusten on havaittu pienentävän lopputuotteeseen jäävien vikojen määrää. Tarkastuksessa vaihetuote käydään läpi ja virheet kirjataan. Katselmus suoritetaan vaiheen päätteeksi, jolloin vaiheen lopullinen tuote käydään läpi ja todetaan vaihe päättyneeksi. Tarkastusten avulla projektin etenemistä on helpompi seurata ja projekti saa luontevia välitavoitteita eli etappeja. (sama, 49-50.)

Medanets ABS -järjestelmän integraatiossa tarkastukset ovat viikkotapaamisia, joissa kasvotusten käydään läpi projektin vaihe ja aikaa on varattu huomioihin sekä kysymyksiin. Viikkoraportti tulee toimittaa päivää ennen tarkastusta muiden luettavaksi. Jokaisesta viikkoraportista kirjoitetaan muistio myöhempää tarkastelua varten. Katselmus tapahtuu jokaisen vaiheen päätteeksi ja katselmoinnissa käydään läpi vaiheessa syntyneet tuotteet. Kaksi päivää ennen katselmusta katselmoitavat tuotteet sekä tyhjä katselmuslomake lähetetään muiden luettavaksi ja lomakkeeseen kirjataan huomionkohteet. Ennen katselmusta lomakkeet palautetaan ja huomionkohteisiin annetaan vastaukset. Katselmuslomakkeet myös käsitellään katselmoinnissa ja lomakkeista kootaan muistio myöhempää tarkastelua varten.

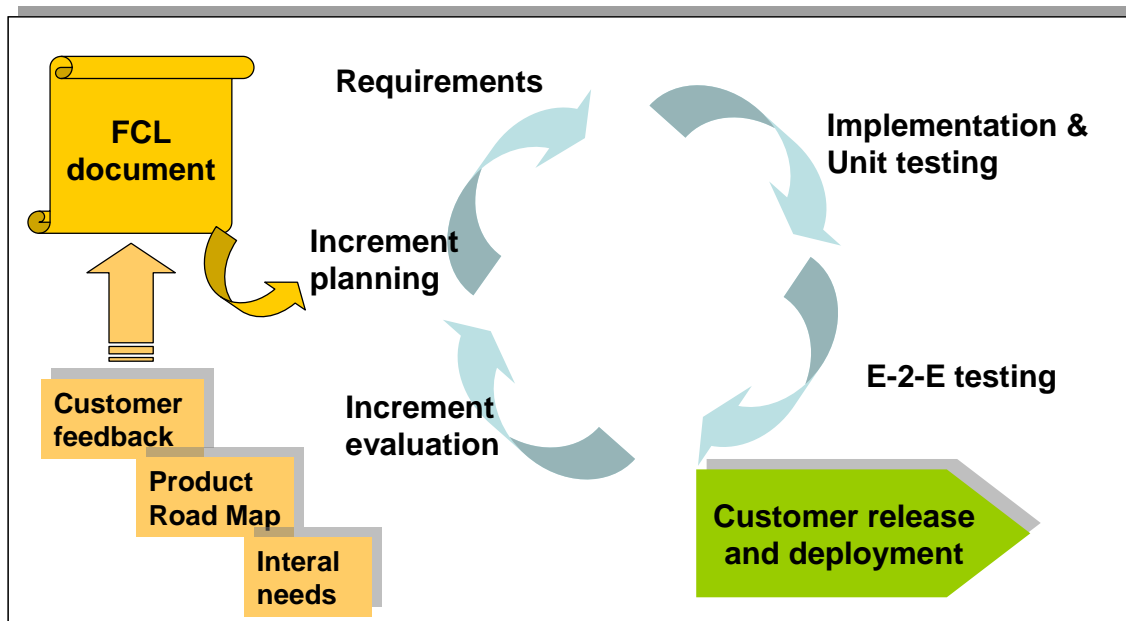


KUVIO 12. Ohjelmistoprojektin katselmukset ja tarkastukset (muokattu) (Ohjelmistotuotanto, Haikala & Märijärvi 2003, 50)

7.2 Ohjelmiston vaihejakomalli

Vaihejakomallilla kuvataan ohjelmiston kehitystyö tai koko elinkaari jakamalla se vaiheisiin. Ohjelmiston elinkaari on aika kehityksen aloituksesta ohjelmiston poistamiseen käytöstä. Tavallisin vaihejakomalli on niin sanottu vesiputousmalli, joka on kuvattu liitteessä 4. Vesiputousmalli yksinään ei ole ohjelmistoprojektissa käytännöllinen, koska asiakasvaatimukset muuttuvat melkein aina projektin edetessä ja vesiputousmallissa edetään päättäväisesti vaihe vaiheelta loppua kohden. Alati muuttuviin vaatimuksiin reagoiminen vaatii kehittyneemmän vaihejakomallin. (Haikala & Märijärvi 2003, 36.)

Työssä käytin Medanets Oy:n tuotanto- ja kehitysprosessin mukaista inkrementaalista vaihejakomallia. Kuviossa 13 on inkrementaaliprosessi kuvattu uuden ominaisuuden tai muutoksen tarpeesta aina julkaisuun saakka. Aluksi ominaisuus, joka on syntynyt joko ominaisuuden tarpeesta, on sisällytetty tuotteen road mappiin tai tulee asiakaspalautteena, siirretään Feature Candidate Listiin (FCL). Listasta potentiaaliset uudet ominaisuudet sisällytetään seuraavaan inkrementtikierrokseen tärkeysasteen mukaan ja inkrementtiin valituille ominaisuuksille tehdään tarkennettu vaatimusmäärittely ennen toteutusta. Inkrementtikierroksen jälkeen, kun ominaisuus on suunniteltu, tuotettu ja testattu, ominaisuus siirtyy valmiiseen julkaisuun asiakkaalle.



KUVIO 13. Medanets Oy inkrementaaliprosessin kuvaus (Medanets Oy 2010d, Tuotekehitysprosessikuvaus v1.0)

Ohjelmistoprojektin lopputuotetta kehitetään pienin inkrementtein eli julkaisuina. Pienistä julkaisuista koostuu lopullinen toimiva kokonaisuus. Etuna inkrementeissä on, että toimeksiantaja näkee työn tuloksia jo ennen lopputulosta ja ehtii antaa muutosehdotuksia ennen kuin on liian myöhäistä. Inkrementaalinen vaihejakomalli sopii laadunvarmistuksen tarkastuksien kanssa yhteen ja tuotoksien julkaisemisesta tulee säännöllistä. (Haikala & Märijärvi 2003, 42; Poimala, Heikniemi & Blåfield, hakupäivä 20.10.2010.)

”Inkrementaalisia menetelmiä on erilaisia ja niissä on suuria eroja sen suhteen, mitä kaikkea projektissa tehdään inkrementaalisesti” (Poimala ym., hakupäivä 20.10.2010). Tässä työssä määrit-

telyn, suunnittelun, toteutuksen ja testauksen olen suorittanut inkrementaalisesti. Osa-alueet koostuvat useista peräkkäin olevista vesiputousmalleista muodostaen toiminnallisen kokonaisuuden. Kun osa-alue on hyväksytty ja katselmoitu, siirrytään seuraavaan ohjelmistoprojektin osa-alueeseen. Riskinä inkrementaalisessa vaihejakomallissa on, että projektin aikana tuleviin muutoksiin voi olla hankala reagoida tai inkrementtien liittämässä voi seurata yllättäviä ongelmia. (Poimala ym., hakupäivä 20.10.2010.)

8 JOHTOPÄÄTÖKSET

8.1 Kehitystehtävän yhteenveto

Continua Health Alliancen standardien käyttö ei ole aivan yksinkertaista. Standardit asettavat tarkat vaatimukset mittauslaitteille ja varsinkin hallintalaitteille. Hallintalaitteen täytyy tukea Continuan hyväksymiä tiedonsiirtoprotokollia ja lääketieteellisiä tiedon tallennusmuotoja. Standardidokumentit ovat kymmeniä tai jopa satoja sivuja pitkiä kuvauksia, säädöksiä ja rajoituksia liittyen standardien käyttöön. Dokumentit myös viittaavat toisiinsa useassa kohtaa vaatien lukijan tutustuvan useaan dokumenttiin yhtä aikaa.

Aiheina Continua Health Alliancen ratkaisujen käyttö ja langaton sairaala ovat tuoreita ja ajankohtaisia. Nykyiset langattomat ratkaisut esimerkiksi sairaaloissa eivät ole suoraan yhteensopivia ja vaativat sairaalan IT-henkilökunnalta tietoa ja taitoa. Yhteensopivien langattomien ratkaisujen malli on edelleen kehitteillä ja on ottamassa jalansijaa vähitellen sairaalaympäristöissä.

Continuan sertifioimat mittauslaitteet eivät täysin sovellu käytettäväksi sairaala- tai terveyskeskuympäristöissä, vaikka Health Device Profile on suunniteltu toimimaan myös sairaaloissa. Mittauslaitteet on suunniteltu kotikäyttöön, mikä asettaa rajoituksia hallintalaitteen ja mittauslaitteen toimintaan. Mittauslaitteet pystyvät muodostamaan yhteyden ainoastaan yhteen hallintalaitteeseen kerrallaan, jolloin muiden hallintalaitteiden käyttö vaatii mittauslaitteen resetoinnin hallintalaitetta vaihtaessa. Resetointi aiheuttaa lisää ajan käyttöä ja vaikeuttaa laitteiden käyttöä. Continua Health Alliance tekee yhteistyötä Integrating Healthcare Enterprisesin kanssa, joten tulevaisuudessa Continuan sertifioimat tuotteet voivat soveltua sairaalakäyttöön paremmin.

Continuan standardien integroiminen valmiiseen tai kehitteillä olevaan ympäristöön on mahdollista. Medanets ABS -järjestelmä on suunniteltu kehitystä silmällä pitäen ja järjestelmään Continuan tarjoamia sovelluksia pystyy lisäämään. Eri mittauslaitetyyppien käyttämät ohjelmistorajapintojen erot ovat pieniä ja eivät vaadi paljon muokkausta liitettäessä erilaista mittauslaitetyyppiä järjestelmään.

Tämän työn kehitystehtävä, jossa Medanets ABS -järjestelmään haluttiin lisätä tuki Continua-standardia noudattaville mittalaitteille, onnistui. Mittaustiedot verenpainemittarilta välittyvät virheettömästi ja langattomasti järjestelmään ja sitä kautta tietokantaan luettaviksi. Integraation käyttö järjestelmässä ei ole ajankohtainen, ennen kuin mittauslaitteet tukevat useampaa hallintalaitetta. Medanets Oy sai kattavan tietopaketin Continuan Health Alliancen tuotteista ja ratkaisuisista tämän opinnäytetyön myötä.

8.2 Jatkokehitys

Työn myötä syntynyt ohjelmisto antaa mahdollisuuden kehittää integraatiota eteenpäin toimimaan myös muiden Continuan mittauslaitetyyppien kanssa. Tällä hetkellä mittauslaitespesifi rajapinta on ohjelmoitu vain verenpainemittarille. Kehityksen myötä myös välittäjälaitteen poistaminen ratkaisusta on tärkeää, koska se helpottaa laitteiden käyttöä ja Continua-standardin integrointiin tarvitaan tällöin vain mittaus- ja hallintalaitte.

CHA:n versio 2010:n suuntaviivat ovat jäsenille saatavilla vuonna 2010 ja muille ostettavissa 2011. Uusien suuntaviivojen myötä Continua tukee myös muita mittauslaitetyyppejä ja tiedonsiirtoprotokollia. 2010 suuntaviivojen käyttöönotolla tässä työssä kehitetty ohjelmisto pysyy ajan tasalla ja ohjelmistoa pystyy kehittämään entistä monipuolisemmaksi.

Jatkossa tulee myös seurata mittauslaitteiden kehitystä ja soveltuvuutta Medanets ABS -järjestelmään. Uusilla suuntaviivoilla toteutetut laitteet voivat mahdollistaa paremmin useamman hallintalaitteen käytön.

LÄHTEET

Kirja

Haikala, I. & Märijärvi, J. 2003. Ohjelmistotuotanto. 9. painos Helsinki: Talentum

Digitaaliset lähteet

Bogia, D. 2007. ISO/IEEE 11073 Personal Health Data Tutorial. Hakupäivä 30.9.2010.
<http://www.lampreynetworks.com/assets/documents/2007-12-21-IEEE-PHD-tutorial.pdf>

Continua Health Alliance 2009. Continua Certification version 1.0. Hakupäivä 11.10.2010.
http://www.continuaalliance.org/static/cms_workspace/Continua_Certification_Public.pdf

Continua Health Alliance 2010a. Join. Hakupäivä 10.10.2010.
<http://www.continuaalliance.org/about-the-alliance/join.html>

Continua Health Alliance 2010b. Mission And Objectives. Hakupäivä 30.9.2010.
<http://www.continuaalliance.org/about-the-alliance/mission-and-objectives.html>

Havlát, M., Morsing, A., Mancardi, F. & Pollans, W. 2008. User Manual TestLink version 1.7.
Hakupäivä 22.10.2010. http://www.teamst.org/_tldoc/1.7/user_manual.pdf

Hughes, R. 2009. HEALTH DEVICE PROFILE Implementation Guidance Whitepaper.
Hakupäivä 30.9.2010.
http://www.bluetooth.com/Research%20and%20White%20Papers/HDP_Implementation_WP_V10.pdf

Küng, S., Onken, L. & Large, S. TortoiseSVN Subversion-käyttöliittymä Windows-ympäristöön
Versio 1.6.11. Suom. Kari Granö. Hakupäivä 20.10.2010.
<http://downloads.sourceforge.net/tortoisesvn/TortoiseSVN-1.6.11-fi.pdf?download>

Medanets Oy 2007. Medanets järjestelmän esite. Hakupäivä 7.10.2010.

http://www.medanets.com/esitteet/Medanets_jarjestelman_esite.pdf

Medanets Oy 2010a. Tuotteet. Hakupäivä 30.9.2010. <http://www.medanets.com/?sivu=tuotteet>

Medanets Oy 2010b. Tuotteet lisätietoja. Hakupäivä 30.9.2010.

http://www.medanets.com/?sivu=tuotteet_lisaa&lisa=1

Medanets Oy 2010c. Langaton sairaala. Hakupäivä 30.9.2010.

<http://www.medanets.com/?sivu=langattomuus>

Medanets Oy 2010d. Tuotekehitysprosessikuvaus v1.0. Sisäinen lähde.

Microsoft Corporation 2008. Visual Studio 2008 Product Comparison. Hakupäivä 22.10.2010.

<http://download.microsoft.com/download/5/f/e/5feb6914-bcdf-432f-81c7-e386812b086a/VisualStudio2008-ProductComparison-v1.08.pdf>

Pakarinen, V. & Pärkkä, J. 2008. Teknologia katsaus Continuassa kehitettävistä standardeista.

Hakupäivä 30.9.2010. <http://virtual.vtt.fi/virtual/hl7/Continua-raportti-julkinen.pdf>.

Poimala, S., Heikniemi, J. & Blåfield, H. Ketterät käytännöt – Iteraatiot ja inkrementit. Hakupäivä

20.10.2010. <http://www.ketteratkatayannot.fi/Ketteryys/IteraatiotJaInkrementit/>

Technical Committee ISO/TC 215 "Health informatics" 2009. Health informatics - Point-of-care medical device communication - Part 20601: Application profile - Optimized exchange protocol.

Sisäinen lähde.

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=54331

The MantisBT Team 2010. Mantis Bug Tracker Administration Guide. Hakupäivä 25.10.2010.

http://docs.mantisbt.org/master/en/administration_guide.pdf

Ott, L. 2010. The Evolution of Bluetooth in Wireless Medical Devices. Hakupäivä 9.10.2010.

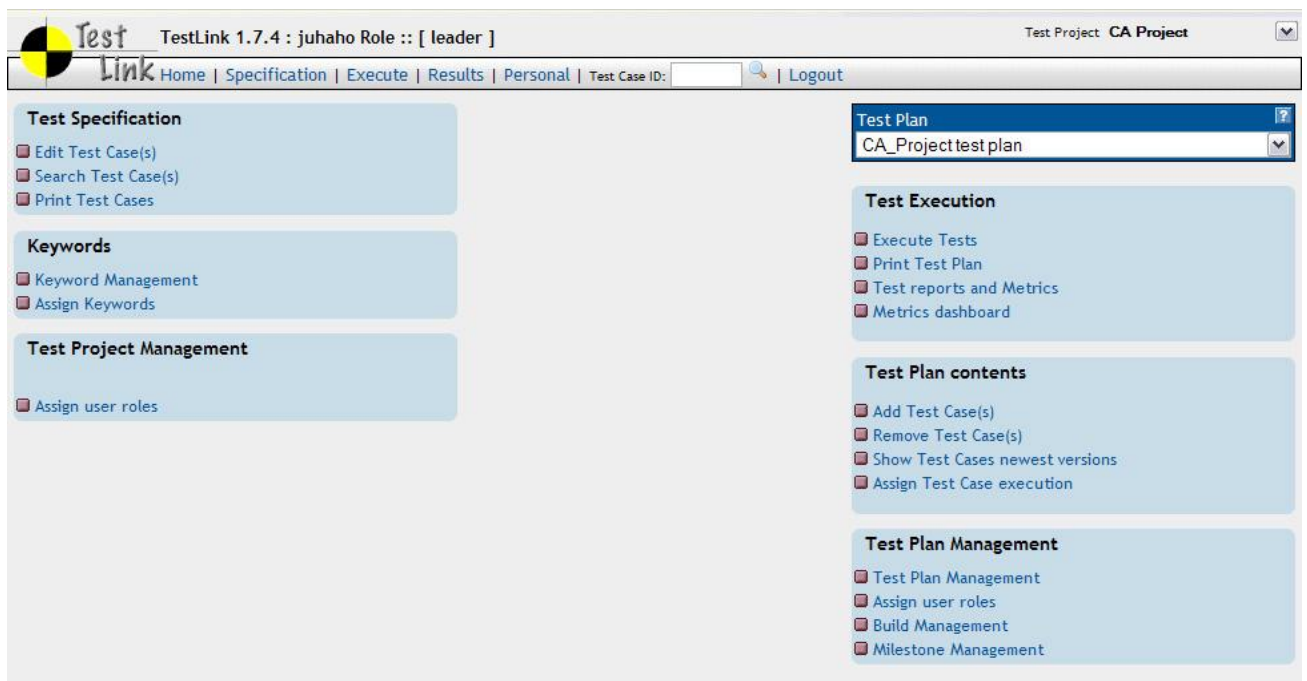
http://www.socketmobile.com/pdf/techbrief/socket_bluetooth-medical_white-paper.pdf.

LIITTEET

LIITE 1

0x00 | 0x00 = obj-handle = 0 (MDS object)
0xFF | 0xFF | 0xFF | 0xFF = event-time (set to 0xFFFFFFFF if RelativeTime is not supported)
0x0D | 0x1D = event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00 | 0x24 = event-info.length = 52
0xF0 | 0x00 = ScanReportInfoFixed.data-req-id = 0xF000
0x00 | 0x00 = ScanReportInfoFixed.scan-report-no = 0
0x00 | 0x02 = ScanReportInfoFixed.obs-scan-fixed.count = 2
0x00 | 0x2C = ScanReportInfoFixed.obs-scan-fixed.length = 44
0x00 | 0x01 = ScanReportInfoFixed.obs-scan-fixed.value[0].obj-handle = 1
0x00 | 0x12 = ScanReportInfoFixed.obs-scan-fixed.value[0].obs-val-data.length = 18
0x00 | 0x03 = Compound Object count (3 entries)
0x00 | 0x06 = Compound Object length (6 bytes)
0x00 | 0x86 = Systolic = 134
0x00 | 0x4E = Diastolic = 78
0x00 | 0x59 = MAP = 89
0x20 | 0x10 | 0x07 | 0x09 = Absolute-Time-Stamp = 2010-07-09
0x13 | 0x35 | 0x08 | 0x00 = Absolute-Time-Stamp = T13:35:0800
0x00 | 0x01 = ScanReportInfoFixed.obs-scan-fixed.value[1].obj-handle = 1
0x00 | 0x12 = ScanReportInfoFixed.obs-scan-fixed.value[1].obs-val-data.length = 18
0x00 | 0x03 = Compound Object count (3 entries)
0x00 | 0x06 = Compound Object length (6 bytes)
0x00 | 0x90 = Systolic = 144
0x00 | 0x46 = Diastolic = 70
0x00 | 0x63 = MAP = 99
0x20 | 0x10 | 0x07 | 0x09
0x13 | 0x35 | 0x54 | 0x00

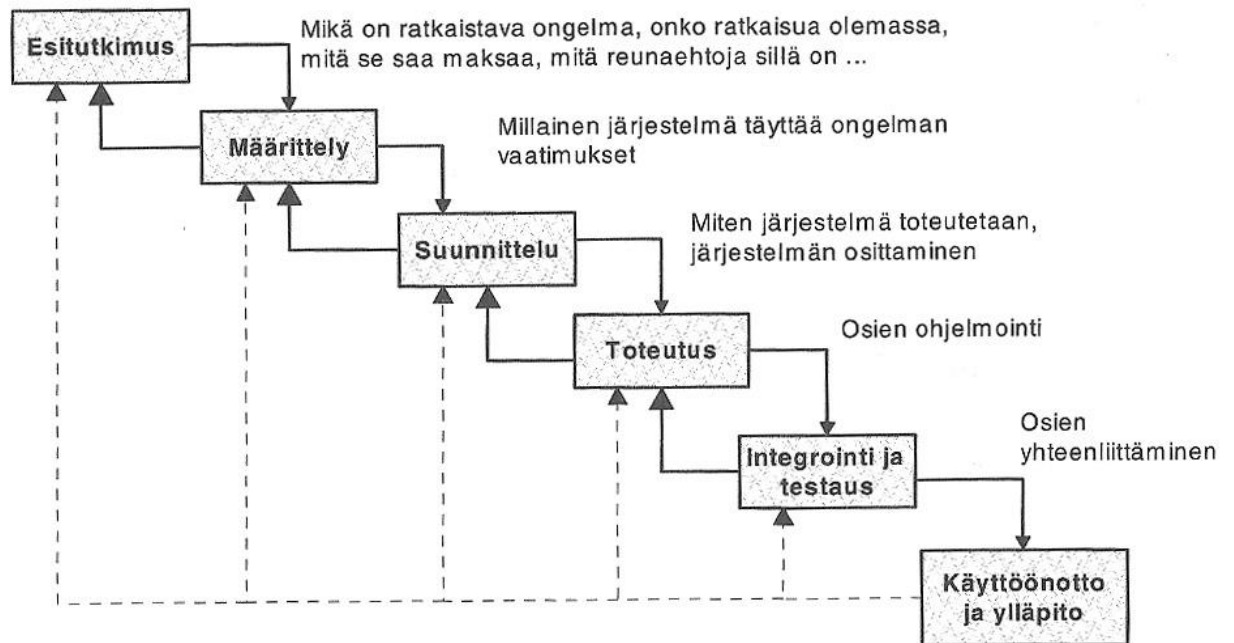
Verenpainemittarin lähettämät kaksi mittaukset heksadesimaalimuodossa ja selitykset luvuille



TestLink-ohjelmiston käyttöliittymä



Prosa-ohjelmistolla tehty yksinkertaistettu sekvenssikaavio mittauslaitteen ja hallintalaitteen välisestä kommunikaatiosta



Esimerkki vesiputousmallista (Ohjelmistotuotanto, Haikala & Märijärvi 2003, 36)