



TAMPEREEN TEKNILLINEN YLIOPISTO

ISMO KARJALAINEN
PÄÄTELAITE- JA PAIKKATIEDON AUTOMATISOITU
KERÄÄMINEN VERKKOPALVELUISSA

Diplomityö

Tarkastajat: professori Mikko Valkama
 tutkija Jukka Huhtamäki

Tarkastajat ja aihe hyväksytty
Tieto- ja sähkötekniikan tiedekunta-
neuvoston kokouksessa 9.12.2009

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Signaalinkäsittelyn ja tietoliikennetekniikan koulutusohjelma

KARJALAINEN, ISMO: Päätelaite- ja paikkatiedon automatisoitu kerääminen verkkopalveluissa

Diplomityö, 94 sivua, 9 liitesivua

Kesäkuu 2010

Pääaine: Tietoliikenneverkot ja protokollat

Tarkastajat: professori Mikko Valkama, tutkija Jukka Huhtamäki

Avainsanat: Automatisoitu tiedonkeruu, päätelaitetieto, paikkatieto, rikas informaatio, käyttökokemus, Drupal, mobiililaitte

Verkkosivustot ovat keränneet jo pitkään tietoja käyttäjistä, mutta tiedot ovat usein rajoittuneet vain selaintietoihin. Muutamina viime vuosina on alettu kerätä tietoa varsinkin mobiililaitteista, sillä yhä useammassa niistä on webselain. Päätelaitetietojen avulla on mahdollista esimerkiksi tehdä uudelleenohjaus verkkosivuston mobiiliversioon tai muokata sivuston ulkoasua päätelaitteen koon mukaan. Tämä parantaa verkkosivuston käytettävyyttä ja saavutettavuutta varsinkin mobiililaitteilla. Lisäksi päätelaitetietoja keräämällä saadaan tietoa siitä, millaisilla laitteilla ja mitä verkkosivuja niillä käytetään. Samoin erilaiset paikkatietoa hyödyntävät palvelut ja sovellukset ovat kasvattaneet suosiotaan viime vuosina. Paikannuksen odotetaan lyövän entistä paremmin läpi lähivuosina etenkin mobiililaitteissa, joissa GPS-vastaanottimien määrä on lisääntynyt vähitellen. Paikkatietoa voidaan hyödyntää monilla toimialoilla, kuten sosiaalisissa verkkopalveluissa, mainostamisessa, logistikkassa ja kuljetuksessa sekä yleisesti teollisuudessa.

Tämän työn tavoitteena on selvittää, mitä eri vaihtoehtoja on automatisoidulle päätelaitetunnistukselle ja paikkatiedonkeräämiselle. Työ on tehty osana Hypermedialaboratorion Urban Computing Lab eli UCL-hanketta, jonka laboratorioympäristö on taustalla motivoimassa tätä tutkimusta. Tutkimuksessa esitellään ne menetelmät, joiden toiminnallisuudesta ja soveltuvuudesta otettiin selvää osana UCL:n toteutusteknologioiden valintaprosessia. Tutkimuksen tuloksena on selvitetty, mitä päätelaitetietoja ja miten niitä voidaan kerätä automatisoidusti verkkopalveluissa. Samoin se osoittaa, mitä keinoja on saada paikkatieto talteen eri päätelaitteita käytettäessä. Tutkimuksen perusteella voidaan todeta, että eri teknologioita yhdistellessä on mahdollista kerätä käyttötietoja, kuten päätelaitteen näytön koko, selain, käyttöjärjestelmä ja sijainti. Tällä työllä on luotu pohja automatisoidun käyttötiedonkeruun toteuttamiselle verkkopalveluissa. Toteutuksen ohessa syntyi myös jatkokehitysideoita UCL:ään ja näkemyksiä tulevaisuudesta liittyen mobiiliteknologioihin ja paikannukseen. Niinpä tämän työn tuloksia tulee vertailla uusiin teknologioihin, joita kehitellään jatkuvasti niin päätelaitetunnistuksen kuin paikannuksen osalta.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Signal Processing and Communications Engineering

KARJALAINEN, ISMO: Automatic device and location detection in

web services

Master of Science Thesis, 94 pages, 9 Appendix pages

June 2010

Major: Communication Networks and Protocols

Examiners: Professor Mikko Valkama, Researcher Jukka Huhtamäki

Keywords: Automatic data acquisition, device information, location information, rich information, use experience, Drupal, mobile device

Web services have been collecting data about their users for a while but the information has often been limited to browser data. During the last couple of years, web services have begun to collect information about mobile devices as more and more of them has a web browser. The device information can be used for example to redirect to the mobile version of the website or modify the site's appearance according to the size of the device. This improves the availability and the accessibility of the website especially with the mobile devices. By collecting the device data, it is possible to get information about what kind of devices and which websites are used. Similarly, various services and applications that use geographical information have increased their popularity in the recent years. It is expected that in the coming years locationing will have its breakthrough since the amount of GPS receivers in mobile devices has risen gradually. Location information can be used in several branches of industry such as social web services, advertising, logistics and transport.

The aim of this thesis is to find out which options there are for automatic device detection and location information collection. This thesis has been done as a part of Hypermedia Laboratory's Urban Computing Lab (UCL) project whose laboratory environment is motivating this research. The research shows several methods whose functionality and applicability were found out during the UCL's development process. As a result this research introduces which device informations are possible to get and how they can be collected as automatically as possible in web services. It also shows what kinds of methods there are to get location information from different devices. The research points out that by combining different technologies it is possible to collect usage related information such as the screen size of the device, browser, operating system and location. This thesis forms a basis for implementing an automatic collector of usage related information in web services. There also rose new ideas to improve UCL and visions about the future related to mobile technologies and locationing. Therefore it is necessary to compare these results to the new technologies that are continuously being developed.

ALKUSANAT

Tämä diplomityö on tehty Tampereen Teknillisen Yliopiston matematiikan laitoksen Hypermedialaboratoriossa, jossa työ liittyi Urban Computing Lab -hankkeeseen. Niin itse projektissa kuin tässä diplomityössäni sain hyödyntää omia vahvuuksiani tiedonhaussa, uusien asioiden omaksumisessa ja projektin teknillisessä toteutuksessa. Haluan kiittää koko Hypermedialaboratoriota mahdollisuudesta työskennellä UCL-projektissa, jossa opin todella paljon uutta mm. päätelaitetiedoista, paikannuksesta, mobiililaitteista, Drupalista ja tutkimuksesta sekä tieteellisestä kirjoittamisesta. Haluan erityisesti kiittää ohjaajiani tutkija Jukka Huhtamäkeä ja professori Mikko Valkamaa mielenkiinnosta työni aihetta kohtaan sekä asiantuntevista kommentteista ja diplomityön joustavasta tarkastamisaikataulusta. UCL-projektin osalta kiitokset ansaitsevat Jukan lisäksi tutkija Jaakko Salonen projektiin perehdyttämisestä ja projektin ohjauksesta sekä tutkimusapulainen Joonas Meriläinen korvaamattomista Drupal-taidoista ja toteutuksen ideoinneista. Lisäksi kiitän erikoistutkija Kirsi Siliusta ja tutkija Anne-Maritta Tervakaria projektissa ja diplomityöseminaareissa saamistani vinkeistä. Anne-Maritalle vielä erityiskiitokset tämän työn lähdeviittauksiin liittyvistä opastuksista. LaTeX-vinkeistä haluan kiittää tutkimusapulainen Teemo Tebestiä ja signaalinkäsittelyn diplomi-insinööri Joonas Nikusta. Hypermedialaboratorion miellyttävän työympäristön ansiosta projekti eteni sujuvasti ja sain työskennellä omaan tahtiini.

Työn ulkopuolelta kihlattuni Satun lisäksi ylivoimaisesti tärkein tuki opiskeluilleni on tullut vanhemmiltani Anterolta ja Anna Liisalta. Myös veljeni Olli ja Ilkka sekä muut sukulaiset ja tuttavat ovat aina kannustaneet opinnoissani ja etenkin niiden ulkopuolisissa asioissa, unohtamatta kaikkia opiskelukavereitani.

Tampereella 6.5.2010

Ismo Karjalainen

SISÄLLYS

1. Johdanto	1
2. Taustaa	3
3. Laitteistotiedonkeruu	7
3.1 Tietojen kerääminen ja siitä saatava hyöty	7
3.2 Selaintunnistus	8
3.2.1 \$_SERVER-muuttuja	8
3.2.2 get_browser()-funktio	9
3.2.3 Opera Mini -selaimen tunnistus	10
3.3 User Agent Profile	12
3.4 Laitteistotiedonkeruukirjastot	14
3.4.1 DetectRight	14
3.4.2 Handset Detection	16
3.4.3 DeviceAtlas	18
3.4.4 Wireless Universal Resource File	19
3.4.5 Mobile Device Information	22
3.5 Laitteistotiedonkeruumenetelmien vertailu	23
3.6 Laitetunnistuksen toteuttaminen Drupalissa	25
3.6.1 Drupal Mobile Plugin	25
3.6.2 Drupal Browscap	26
4. Paikannus	27
4.1 Paikannuksen hyödyt ja ongelmakohdat	27
4.2 Paikannusvaihtoehtoja	29
4.3 Verkkopaikannus	29
4.4 Lähipaikannus	32
4.5 Satelliittipaikannus	35
4.5.1 Global Positioning System	36
4.5.2 Assisted GPS	38
4.5.3 Galileo-paikannusjärjestelmä	40
4.6 Solupaikannuskirjastot	42
4.6.1 OpenCellID	42
4.6.2 CellID Novay	44
4.6.3 CellSpotting	45
4.7 Geolocation API	46
4.8 Paikannusta hyödyntäviä palveluita	47
5. Tapauskuvaus: UCL	51
5.1 UCL-verkkoympäristön esittely	51
5.2 Käyttökokemuksen tallentaminen	53

5.3	Paikannuksen toteutus	55
5.3.1	Java ME	56
5.3.2	Python for S60	58
5.3.3	Sovellusesimerkkejä	59
5.4	Päätelaitetunnistuksen toteutus	61
5.4.1	Teeman valinta	62
5.4.2	Laitetietojen kerääminen käyttökokemuslomakkeelle	62
5.4.3	Laitetunnistuksen tuloksia	66
5.5	Testiympäristöt	69
5.5.1	Remote Device Access Services	70
5.5.2	Opera Mini -simulaattori	73
5.5.3	User Agent Switcher	74
5.6	Käyttäjien palaute	75
6.	Pohdintaa	77
6.1	Päätelaite- ja paikkatiedonkeruun onnistuminen	77
6.2	Jatkokehitysideoita	78
6.3	Tulevaisuuden näkymiä	80
7.	Yhteenveto	83
	Lähteet	93
Liite1:	Mobile Plugin -esimerkkejä	95
Liite2:	Browscap-esimerkkejä	97
Liite3:	Päätelaitetunnistuksen tuloksia UCL:ssä	99

KUVAT

3.1	Opera Minin välityspalvelimet prosessoivat sisällön uudelleen pääte- laitteen ja webpalvelimen välillä (Bandeekar, 2005)	10
3.2	Handset Detectionin toimintaperiaate (Handset Detection, s.a.-b) . .	17
3.3	DeviceAtlaksen toimintaperiaate (DeviceAtlas, s.a.-a)	19
3.4	mDevInfin käyttöliittymä (McLachlan, s.a.-a)	23
4.1	Paikannusmenetelmien toimintaympäristö ja tarkkuus (Rainio, 2003, s. 5)	30
4.2	IP Location Finderin paikannustulos Google Mapsin näkymässä . . .	35
4.3	A-GPS:n toimintaperiaate (Djuknic & Richton, 2001)	39
4.4	OpenCellID:n tukiasemasolut Tampereen keskustassa	43
4.5	CellSpottingin tukiasemasolut Tampereen keskustassa	45
5.1	UCL-verkkoympäristön etusivu N73:ssa Nokian ja Opera Minin selai- messä	52
5.2	UCL-verkkoympäristön etusivu Firefox-selaimessa	53
5.3	Whereis-moduulin paikannustulos Goole Mapsin kartalla	60
5.4	MyLocationin paikannustulos Nokia N97:ssa	61
5.5	RDA:n toimintaperiaate (Forum Nokia Wiki, 2009d)	71
5.6	RDA:n selainnäkyvä käyttäjille	71
5.7	RDA:n käyttöliittymä - puhelimenä Nokia N73	72

TAULUKOT

3.1 Laitteistotiedonkeruukirjastojen tärkeimmät ominaisuudet	24
--	----

LISTAUKSET

3.1	Esimerkki \$_SERVER-muuttujan käytöstä (W3Schools, s.a.-b; W3Schools, s.a.-a)	8
3.2	Esimerkki get_browser()-funktion käytöstä (W3Schools, s.a.-b)	9
3.3	Esimerkki HTTP- ja Opera Minin HTTP-otsikoiden käytöstä	11
3.4	Ote Nokia 6230i:n laitteistotiedoista (Developer's Home, s.a.)	13
3.5	DetectRightin PHP-rajapinnan käyttö	15
3.6	Handset Detectionin PHP-rajapinnan käyttö	17
3.7	WURFL:n DTD (Openwave Developer Network, s.a.)	21
3.8	Nokia N73 WURFL:n XML-dokumentissa	21
3.9	WURFL:n PHP-rajapinnan käyttö (Forum Nokia Wiki, 2009a)	22
4.1	OpenCellID:n koordinaatit XML-dokumentissa solunumerolle 22440	43
4.2	CellID Novayn koordinaatit haetuilla tiedoilla (maakoodi eli MCC 224, operaattorikoodi eli MNC 5, aluekoodi eli LAC 29121, solun numero eli CID 22440)	44
5.1	Sovelluskirjanmerkin koodi työpöytäselaimille	55
5.2	Sovelluskirjanmerkin koodi matkapuhelinten selaimille	55
5.3	Selaintunnistukseen käytetyt säännöt, joilla valitaan oikea tyylitiedosto	62
5.4	Jokaisesta laitteesta haetaan perustiedot eri menetelmin	63
5.5	Lomakkeelle haetaan työpöytäselaimen tapauksessa tiedot Browserin avulla	64
5.6	Menetelmät joilla mobiililaitteen tiedot kerätään	65
5.7	Opera Minin demon tiedot haetaan WURFL:lla	66
5.8	Nokia N96:n selaimen kuvaus RDF-dokumentissa	68
5.9	Nokia E70:n selaimen kuvaus RDF-dokumentissa	68
5.10	Esimerkki User Agent Switcherin käytöstä	74

TERMIT JA SYMBOLIT

A-GPS	Assisted Global Positioning System
AOA	Angle of Arrival
ASCII	American Standard Code for Information Interchange
API	Application Programming Interface
CC/PP	Composite Capabilities/Preferences Profile
CI	Cell Identification
CSS	Cascading Style Sheets
CSV	Comma-Separated Value
DB	Database
DTD	Document Type Definition
EOTD	Enhanced Observed Time Difference
FAQ	Frequently Asked Questions
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
ID	Identification
IP	Internet Protocol
IPDL	Idle Periodic Down Link
JS	JavaScript
LAN	Local Area Network
LF	Location Fingerprint
LTE	Long Term Evolution
MMS	Multimedia Messaging Service
OEM	Original Equipment Manufacturer
OTDOA	Observed Time Difference of Arrival
PHP	Hypertext Preprocessor
RDA	Remote Device Access
RDF	Resource Description Framework
RFID	Radio-frequency Identification
Rx	Signal Level
SIM	Subscriber Identity Module
TA	Timing Advance
TCP/IP	Transmission Control Protocol / Internet Protocol
TDOA	Time Difference of Arrival
TOA	Time of Arrival
UA	User Agent

UAProf	User Agent Profile
UCL	Urban Computing Lab
URL	Uniform Resource Locator
VDL	Virtual Developer Lab
WALL	Wireless Abstraction Library
WAP	Wireless Access Protocol
WLAN	Wireless Local Area Network
WNG	WALL Next Generation
WURFL	Wireless Universal Resource File
XML	Extensible Markup Language
XML-RPC	XML Remote Procedure Call
XSLT	Extensible Stylesheet Language Transformations

1. JOHDANTO

Tutkimuksen tarkoituksena on selvittää, mitä kontekstitietoja on mahdollista kerätä päätelaitteista verkkopalveluissa. Työssä kartoitetaan menetelmiä ja teknologioita erilaisten päätelaite- ja paikkatietojen keräämiseen. Toisin sanoen selvitetään, miten saadaan mahdollisimman automatisoidusti tieto siitä, millainen päätelaite käyttäjällä on ja missä sitä käytetään. Lisäksi työssä käydään läpi syitä siihen, miksi näitä tietoja tulisi kerätä ja mitä sen avulla voidaan saavuttaa. Diplomityön sisällön taustalla on Hypermedialaboratorion hanke *Urban Computing Lab* eli *UCL*, jonka teknisen toteutuksen kartoitukseen työ liittyy. *UCL* on verkkopohjainen laboratorioympäristö, jolla kerätään systemaattisesti informaatiota erilaisten verkkopalveluiden käyttökonteksteista ja niihin liittyvistä eritasoisista tekijöistä. *UCL* on laiteriippumaton ympäristö, jotta mahdollisimman monesta päätelaitteesta saataisiin tarvittavat laitetiedot talteen. Samoin laitteiden kirjo vaikuttaa paikkatiedonkeruun toteutukseen. Tämä työ kartoittaa erilaisia toteutusteknologioita näiden kontekstietojen automatisoituun keräämiseen sekä esittelee, miten teknologioiden käyttöönotto onnistui *UCL*:ssä.

Jukka Huhtamäki on tutkinut osana diplomityötään kuvailutiedon esittämistä, jonka laitteiden ja käyttäjien osalta mahdollistavat *Composite Capabilities/Preferences Profile* (*CC/PP*) ja sen päälle rakennettu *User Agent Profile* (*UAProf*). Tässä työssä *UAProf* on keskeisessä osassa laitetietojen keräämisessä. *UAProf* määrittelee sanaston laitetietojen kuvaamiselle *RDF-dokumentteina*. (vrt. Huhtamäki, 2005) Näiden dokumenttien hyödyntäminen ei kuitenkaan ole optimiratkaisu automatisoidulle käyttötiedonkeruulle, jota palvelevat paremmin keskitetyt *laitteistotiedonkeruukirjastot*. Näistä puolestaan ei löydy tutkimustuloksia, joten tämän työn tarkoitus on antaa pohja näiden päätelaitetietojen keräämiseen tarkoitettujen kirjastojen kartoitukselle. Paikannusvaihtoehtoja mobiililaitteissa on puolestaan esitelty muun muassa Antti Rainio VTT:n teknologiakatsauksessaan (143/2003) *Paikannus mobiilipalveluissa ja sovelluksissa*. Rainio esittelee julkaisussaan yleisesti eri paikannusmenetelmät mobiililaitteille ja mainitsee useita käyttökohteita mobiilipaikannukselle. Myös muita mobiililaitteiden paikannusta käsitteleviä tutkimuksia löytyy paljon. Tämä johtuu siitä, että niin käyttäjien kuin laitteiden ja esineiden paikantamiselle on ilmennyt tarvetta monilla eri aloilla. Ihmiset kuljettavat puhelinta käytännössä aina mukanaan ja logistiikan kannalta on tärkeää tietää, missä tavarat liikkuvat tiettyinä

ajanhetkenä. Merkittävin julkaisu UCL:n osalta on Kirsi Siliuksen ja Anne-Maritta Tervakarin *Learning by doing as mobile with Urban Computing Lab*, johon viitataan tämän työn luvussa 2. Siinä esitellään UCL:n toteutuksen taustoja ja tavoitteita, jotka toimivat motivaationa tähän työhön liittyvälle tutkimukselle.

Tämän työn teoriaosuus jakautuu kahdeksi kokonaisuudeksi, jotka ovat laitteistotiedonkeruu ja paikannus. Laitteistotiedonkeruuosiossa eli luvussa 3 pohditaan, miten ja miksi laitetietoja tulisi kerätä. Tässä motivointina on UCL ja siihen haluttu päätelaitetiedot. Luvun keskeisimpänä aiheena on selvittää, minkälaisia valmiita ja yksinkertaisia ratkaisuja on tarjolla päätelaitetunnistukseen sekä kartoittaa muutamia eri laitteistotiedonkeruukirjastoja, jotka sisältävät mobiililaitteiden tietoja eri muodossa. Tähän tutkimukseen on valittu potentiaalisimmat vaihtoehtoja UCL:n tarpeita ajatellen. Eri vaihtoehtoja vertaillaessa selvitetään, mitä ja miten päätelaitetietoja on saatavilla sekä vertailla vaihtoehtoja keskenään muun muassa laitetietokannan koon ja maksullisuuden osalta. Lisäksi luvussa kartoitetaan eri Drupalin moduuleja, sillä niillä oli merkittävä vaikutus UCL:n toteutusteknologian valinnassa.

Luvussa 4 kerrotaan yleisesti paikannuksesta. Huomio keskittyy eri paikannustekniikoihin, paikannusta hyödyntäviin sovelluksiin ja paikannuksen ongelmakohtiin. Tässä luvussa paikannustekniikoiden esittely keskittyy enimmäkseen lähiverkko- ja satelliittipaikannukseen, jotka olivat keskeisessä liittyen UCL:n toteutusvaihtoehtoihin. Mobiililaitteiden osalta selvitetään A-GPS:n toimintaa ja tukiasemasolujen avulla tehtävää paikannusta. Jälkimmäisen selvityksessä esitellään tukiasemasolukirjastoja, jotka sisältävät tukiasemien sijainteja matkapuhelinverkon avulla tapahtuvaan paikannukseen. Samalla selvitetään, mitä uutta satelliittipaikannusjärjestelmä Galileo tuo valmistuessaan mukanaan ja miten työpöytäselaimella voidaan paikantaa käyttäjä.

Luku 5 sisältää UCL:n toteutuksen esittelyn, joka painottuu teoriaosuuden tapaan päätelaitetunnistuksen ja paikannuksen toteutuksiin. Toteutusten tuloksien lisäksi esitellään, miten eri apuvälinein UCL:ää testattiin. Luvun lopussa esitellään käyttäjiltä saatuja palautteita. Luvussa

Luku 6 sisältää pohdintaa, johon liittyy UCL:ään valittujen toteutusteknologioiden onnistuminen. Lisäksi luvussa esitellään jatkokehitysideoita ja tehdään katsaus tulevaisuuden näkyymiin. Luku 7 on työn yhteenveto, jossa kootaan tutkimuksen tulokset yhteen.

2. TAUSTAA

Työn lähtökohtana on toteuttaa Tampereen teknillisen yliopiston matematiikan laitoksen alaisuudessa toimivalle Hypermedialaboratoriolle verkkosovellusympäristö. Työnimeksi tälle verkkopohjaiselle laboratorioympäristölle annettiin Urban Computing Lab eli UCL. Sen avulla opiskelijat voivat systemaattisesti tallentaa erilaisiin käyttötilanteisiin liittyvää rikasta informaatiota myöhempää analysointia varten. Analysoinnin tuloksia voidaan hyödyntää verkkopalveluiden suunnittelu-, toteutus- ja arviointityössä. Tavoitteena on tehdä ympäristöstä laiteriippumaton, jotta käyttötilanteisiin saataisiin vaihtelevuutta niin käyttöpaikkojen ja -tilanteiden kuin pääteelaiteiden osalta. UCL:n kehittämisessä otetaan alusta alkaen huomioon käytettävyyttä, saavutettavuus ja kontekstiedon automatisoitu kerääminen. Tällä pyritään takaamaan itse UCL:n käytön mahdollisimman laadukas käyttökokemus. Tämän toivotaan motivoivan ja kannustavan opiskelijoita laboratorioympäristön käytössä. Samalla tarkoitetaan pystyä havainnollistamaan konkreettisesti teorioissa esitettyjen asioiden ymmärtämistä ja oppimista sekä mahdollistaa omakohtainen kokeileminen ja tietojen monipuolinen analysointi. (vrt. Silius, Tervakari, & Miilumäki, 2009)

Motivaatio ja tarve tällaisen laboratorioympäristön kehittämiselle on tullut Tampereen teknillisen yliopiston hypermedian opiskelijoiden palautteen perusteella. Vaikka luennoitsijat ovat esittäneet kursseilla paljon käytännön esimerkkejä hypermediaprojekteista teoriaan viitaten, on palautteen mukaan kursseille toivottu enemmän esimerkkejä arkielämän tilanteista. Opetettavia asioita hypermedian kursseilla ovat esimerkiksi käytettävyyttä, sisällöntuotanto ja suunnittelu, rakenteiset dokumentit, verkkopalveluiden käyttökelpoisuus sekä hypermedian ohjelmointi. UCL:n suunnittelu ja toteutus tehdään saadun palautteen perusteella. Sen tarkoitus on tukea verkkopalveluiden suunnittelun ja arvioinnin opetusta ja oppimista. Oppimisen ja opiskelijoiden motivoinnin kannalta on tärkeää, että opetukseen liittyy todenmukaisia ratkottavia ongelmia (*tilanteita*) ja asiayhteyksiä teoriaan (*konteksti*). UCL:ään on tarkoitus kerätä rikasta, ajantasaista tietoa laadukkaista todellisista käyttökokemuksista ja kontekstityötä siitä, missä ja mihin eri tarkoituksiin verkkopalveluita käytetään. Yksi UCL:n tärkeimmistä ominaisuuksista on visualisointityökalu. Se auttaa niin opiskelijoita kuin tutkijoitakin ymmärtämään ja analysoimaan kerättyä dataa. (Silius et al., 2009)

Verkkopalvelun laadukas käyttökokemus on monitasoinen ja moniulotteinen il-

miö. UCL:n visualisoinnin tarkoitus on tukea opiskelijoita ymmärtämään paremmin laadukkaan kokemuksen ulottuvuudet ja niiden suhteet toisiinsa. Tulokset auttavat opiskelijoita ymmärtämään, miten ja miksi eri käyttäjät pitävät tiettyjä verkkopalvelun ominaisuuksia parempina kuin toisia ja miksi käyttäjien palaute laadukkuudesta vaihtelee. On tärkeätä myös muistaa, että tulokset helpottavat suunnittelijoita tekemään priorisoituja valintoja parantaakseen verkkopalvelun laatua. Tästä esimerkkinä voi olla vaikkapa mahdollinen tarve erilliselle, työpöytäkäyttöä kevyemmälle mobiilinäkymälle, jos mobiilikäyttäjiä on paljon. Lisäksi esimerkiksi tieto verkkopalvelua käyttävien päätelaitteiden resoluutioista, käyttöjärjestelmistä ja selaimista helpottaa ulkoasun suunnittelua ja toteutusta. UCL:ssä kerätty tieto voidaan jakaa viiteen eri tasoon: yleinen, fyysinen, sosiaalinen, toiminnallinen ja affektiivinen (Silius et al., 2009):

- Yleinen taso: demograafinen tieto, käyttäjän kokemus yleisesti webin ja verkkopalveluiden käytöstä
- Fyysinen taso: esim. paikkatieto, paikkatietoon liittyvä tägi (käyttäjän itse määrittelemä), aikaleima, mahdollinen valokuva, teknistä tietoa käytettävästä laitteistosta
- Sosiaalinen taso: sosiaalista tilannetta kuvaavaa informaatiota (käyttäjän kuvaus, onko yksin, ystävien kanssa, onko muita ihmisiä läsnä jne.)
- Toiminnan taso: käyttäjän toimintaa kuvaavaa informaatiota (käyttäjän kuvaus, mitä on tekemässä, mitä verkkopalvelua käyttää jne.)
- Affektiivinen taso: käyttäjän tunnetilaa kuvaavaa informaatiota

Konkreettisesti UCL:n on siis tarkoitus olla työväline opiskelijoille viikkotehtäviä ja harjoitustöitä varten. UCL on laiteriippumaton, joten sitä voi käyttää pöytäkoineiden tapaan niin kannettavilla, matkapuhelimilla kuin PDA-laitteillakin. Lisäksi Tampereen kaupunki on mahdollistanut *Langaton Tampere* -hankkeellaan työntekijöilleen sekä Tampereen yliopiston ja Tampereen teknillisen yliopiston opiskelijoille WLAN:n käytön. Nykyisin WLAN:n ja mobiililaitteiden käyttö on yleistynyt esimerkiksi kahviloissa, ostoskeskuksissa, kirjastoissa ja puistoissa. Täten myös kysyntä laiteriippumattomille verkkopalveluille on kasvanut. Niinpä UCL:ään kerätty kontekstitieto on tärkeää, jotta tiedetään, mitä ja missä palveluita käytetään esimerkiksi mobiilisti. Tämä puolestaan takaa totuudenmukaisemman tiedon käytöstä ja koetusta laadusta. (Silius et al., 2009)

Urban Computing Lab -ympäristön avulla voidaan koota nykyistä tehokkaammin rikkaampaa ja reaaliaikaisempaa informaatiota verkkopalvelun käyttötilanteesta. Verkkopohjaisen ja laiteriippumattoman laboratorioympäristön etuja ovat (Silius et al., 2009):

- Käyttötilanteeseen liittyvä informaatio saadaan talteen mahdollisimman "tuoreena", jolloin ei tarvitse turvautua muistikuviin tai usein puutteellisiinkin muistiinpanoihin.
- Informaatio saadaan talteen valmiina digitaalisessa muodossa, joten informaation siirtäminen ja yhdisteleminen muihin informaatioresursseihin, kuten myös erilaisten analyysien ja tiedonlouhintamenetelmien hyödyntäminen, on aiempaa helpompaa.
- Laboratorioympäristön avulla voidaan helposti koota erilaista hyödyllistä informaatiota käyttötilanteesta, jonka kerääminen muutoin olisi kohtuuttoman työlästä. Tällaisia ovat esimerkiksi maantieteellistä sijaintia (paikkatietoa) tai laitteen teknisiä ominaisuuksia koskeva informaatio.
- Opiskelijoiden ei tarvitse olla alan tutkimuksen erityisosajia ymmärtääkseen visuaalisen datan kokonaisuutta, joka kuvastaa erilaisia käyttäjäryhmiä ja käyttötilanteita erilaisten verkkopalveluiden käytössä.

Teknisestä näkökulmasta UCL hyödyntää valintoja mobiilipaikkatiedon ja kontekstitiedon keräämisessä. Jälkikäteen käyttäjät voivat rikastaa tietoja esimerkiksi liittämällä käyttökokemuksiin kuvia, videoita ja kommentteja. Lisäksi UCL:n avulla pyritään löytämään vastauksia sellaisiin kysymyksiin, kuten onko opiskelijoiden välillä eroja, kun käytetään tageja ja huomautuksia, minkälaisilla laitteilla UCL:ää käytetään ja millaisia vaikutuksia kasvavalla mobiilikäytöllä on UCL:n käyttöön. (Silius et al., 2009)

Urban Computin Labia voidaan käyttää opetuksessa, jossa harjoitellaan asiakas- ja käyttäjälähtöistä suunnittelua sekä verkkopalveluiden laadun ja käyttökelpoisuuden arvioimista. Näihin tarvitaan mahdollisimman rikasta ja yksityiskohtaista informaatiota verkkopalveluiden käyttökonteksteista ja niiden eri tasoista. Esimerkkeinä UCL:ää hyödyntävistä kursseista ovat *MATHM-57350 Verkkopalvelun sisällöntuotanto* 5 op, *MATHM-47300 Verkkopalvelun käyttökelpoisuus ja arviointi* 5 op sekä *MATHM-57100 Hypermedian ohjelmointi* 4 op. Näissä laboratorioympäristö voi olla apuna opiskelijoiden viikkoharjoitustehtävien tai harjoitustöiden toteutuksessa silloin, kun esimerkiksi halutaan monipuolisempaa tietoa mobiililaitteilla käytettävien verkkopalveluiden käyttötilanteesta sekä käyttötilanteen eri tasoista. (Silius et al., 2009)

Urban Computing Labin hankehakemuksen sekä keväällä 2009 tehdyn tietomallin ja toimintoprosessien perusteella on selvillä, mitä tietoja käyttäjiltä ja heidän laitteistansa olisi hyvä saada talteen. Tässä vaiheessa epäselvyydet liittyvät lähinnä

siihen, miten kerääminen tulisi toteuttaa automatisoidusti ja mitä tietoja ylipäättän-
sä on mahdollista saada selville. Niinpä alkuun tavoitteena on saada vain mahdol-
lisimman paljon laitetietoja talteen automatisoidulla tiedonkeruulla ja päättelyllä.
Kartoitus käynnistettiin heinäkuussa 2009, jolloin keskeisimpiä kysymyksiä olivat
muun muassa mitkä ovat oleelliset laite-, käyttö- ja yhteisympäristöt eli minkälai-
sia käyttöön liittyviä skenaarioita voidaan ajatella UCL:n tapauksessa olevan. Yksi
keskeisimmistä kysymyksistä alkuvaiheessa oli testilaitteiden saatavuus. Alkuun pi-
tti siis kartoittaa, mistä saataisiin puhelimia ja SIM-kortteja työpöytäkoneiden ja
kannettavien rinnalle testaukseen. Tässä vaiheessa haluttiin myös ottaa selvää, mi-
tä tietoja on mahdollista kerätä tavallisella Web-selaimella ja mitkä ovat erillisten,
puhelimiin asennettavien ohjelmistojen lisäarvot. Erillisten ohjelmien käyttöön suh-
tauduttiin kuitenkin varauksella. Niiden arveltiin lisäävän mobiilikäytön kynnystä,
sillä ohjelmien asentaminen saatetaan kokea hankalaksi ja täten heikentävän sekä
saavutettavuutta että käytettävyyttä.

3. LAITTEISTOTIEDONKERUU

Urban Computing Lab -hanketta varten tehty tietomalli määrittelee, mitä tieto- ja päätelaitteista halutaan saada talteen. UCL:n tapauksessa tällaisia ovat käyttöjärjestelmä, selain ja sen versio, IP-osoite, UAProf-linkki, näytön resoluutio, JavaScript-tuki sekä laitteen tyyppi (työpöytäkone, kannettava vai mobiililaitte). Myös yhteystyypin automaattista tunnistusta pidettiin tarpeellisena tietona. Valmiiden laitetietokantojen lisäksi matkapuhelimeen asennettavien erillisten ohjelmien lisäarvot haluttiin saada selville. Tiedossa nimittäin oli, että puhelimet sisältävät runsaasti hyödyllistä, joskin vaikeasti saatavilla olevaa tietoa. Keskeisin näistä on paikkatieto, josta tarkemmin luvussa 4.

3.1 Tietojen kerääminen ja siitä saatava hyöty

Yksi laitetietojen keräämisen yleisimmistä hyödyistä on *uudelleenohjaus*. Tämä tarkoittaa, että kun käyttäjä avaa websivustoa, hänet voidaan ohjata joko sivuston mobiilipuolelle tai muokata kaikille päätelaitteiden tarkoitetun sivun ulkoasua esimerkiksi tyylitiedostojen avulla. Tunnistuksesta saatuja tietoja, kuten päätelaitteen näytönkokoja, selainta ja sen versiota sekä kosketusnäyttöä, voidaan käyttää apuna, kun muokataan mobiilisivustolle sopivaa ulkoasua. Tällöin näytönkoon ja kosketusnäytön mukaan voidaan vaihtaa tekstien ja kuvien kokoa sekä asettelua. Samoin sivuston *optimoinnista* on hyötyä myös sen lataamisnopeuksien suhteen, sillä yksinkertaisemmat tyylit, pienemmät kuvat ja karsittu sisältö vähentävät tiedonsiirron tarvetta huomattavasti. Kevyet sivustot nopeuttavat selailua, mikä on yleensä tarpeen lähiverkkoa hitaampaa mobiiliyhteyttä käytettäessä. Ilman päätelaitetunnistusta sama sisältö näytetään kaikille päätelaitteille samana, mikä vaikuttaa sivuston käytettävyyteen.

Esimerkiksi Digitoday¹ käyttää sivustollaan päätelaitetunnistusta ja hyödyntää tietoja sisällönmuokkauksessa. Mobiililaitteille tarkoitettu palvelu² tunnistaa matkapuhelimen ja toimittaa sisällön sille oikean kokoisena. Kun palvelu tunnistaa päätelaitteeksi iPhoneen, uudelleenohjaa se käyttäjän palvelun iPhone-versioon³, joka on suunniteltu erityisesti kosketusnäyttöpuhelimia varten. Päätelaitetunnistukseen

¹<http://www.digitoday.fi>

²<http://m.digitoday.fi>

³<http://iphone.digitoday.fi>

Digitoday käyttää mobiilisivustollaan WURFL:ia eli avoimen lähdekoodin päätelaitetunnistusta (Sanoma News, 2008), josta kerrotaan tarkemmin luvussa 3.3.4. Vastaavanlaista uudelleenohjausta ja sisällönmuokkausta käyttävät myös muun muassa Youtube⁴ ja Facebook⁵.

3.2 Selaintunnistus

Laitetietojen keräämiseen löytyy niin sanottuna valmiita ratkaisuja, jotka voidaan ottaa käyttöön PHP:tä käytettäessä. PHP:ssa on kaksi ratkaisua selaintunnistusta varten. Yksi ratkaisu on käyttää globaalia muuttujaa nimeltään `$_SERVER`⁶ ja toinen on valmis funktion nimeltään `get_browser()`⁷. Näiden avulla saadaan HTTP-liikennöintiin ja selaimen liittyviä perustietoja, kuten IP, luvussa 3.3 tarkemmin esitelty *User Agent Profile* eli päätelaitteen profiili, selain ja sen versio sekä käyttöjärjestelmä. Lisäksi valmiit ratkaisut käyvät myös Opera Mini -selaimen, joka hyödyntää välityspalvelimia liikennöintiin ja käyttää omia HTTP-otsikkoja.

3.2.1 `$_SERVER`-muuttuja

PHP:n `$_SERVER`-muuttujan sisään voidaan sijoittaa erilaisia ympäristömuuttujia, jotka palauttavat tietoja esimerkiksi webpalvelimesta ja ylläpitäjistä (NCSA HTTPd, 2008). Erityisen hyödyllinen näistä on `HTTP_USER_AGENT`, joka palauttaa käyttäjän selaimen UAProf-tiedot, jotka voidaan puolestaan säännöllisten lausekkeiden avulla pilkkoa eri osiin (vrt. Babin, Good, Kromann, & Stephens, 2005, s. 467-469). Listauksessa 3.1 on edellä mainitun esimerkin lisäksi muuttuja `REMOTE_ADDR`, joka palauttaa webpalvelimelle pyynnön lähettäneen laitteen IP-osoitteen.

Listaus 3.1: Esimerkki `$_SERVER`-muuttujan käytöstä (W3Schools, s.a.-b; W3Schools, s.a.-a)

```
1 <?php
2 print ( 'UA: ' );
3 echo $_SERVER[ 'HTTP_USER_AGENT' ];
4 print ( '<br />' );
5 print ( 'IP: ' );
6 echo $_SERVER[ 'REMOTE_ADDR' ];
7 ?>
```

⁴<http://www.youtube.com>

⁵<http://www.facebook.com>

⁶<http://www.php.net/manual/en/reserved.variables.server.php>

⁷<http://php.net/manual/en/function.get-browser.php>

Palauttaa esimerkiksi:

```
UA: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
.NET CLR 1.1.4322)
IP: 130.230.x.y
```

3.2.2 get_browser()-funktio

PHP:ssa on myös valmis funktio `get_browser()`, jonka avulla saadaan käyttäjän selaimen tarkemmat tiedot ilman, että tarvitsisi itse käyttää esimerkiksi säännöllisiä lausekkeita. Tämän funktion avulla saadaan muutakin tietoa kuin selaimen versio ja käyttöjärjestelmä, sillä funktio palauttaa muun muassa tuetun CSS-version numeron ja kertoo, onko selaimen evästeet käytössä sekä tukeeko selain JavaScriptiä. (The PHP Group, s.a.) Funktion `get_browser()` käyttöönotto on tosin vaikeampaa kuin edellä mainitun `$_SERVER`-muuttujan, sillä funktiota varten palvelimella täytyy olla selaintietoja sisältävä *browscap.ini*-tiedosto. Tiedosto tulee päivittää itse säännöllisin väliajoin, muutoin ei uusimpia selaimia tunnisteta oikein. (Babin et al., 2005, s. 467) Tiedosto on alun perin Microsoftin tekemä, mutta kukaan yrityksestä ei ole ylläpitänyt sitä, ja tehtävä on jäänyt vapaaehtoisille. Nykyisin tiedostoa ylläpitää Gary Keith, joka on päivittänyt tietoja vuodesta 1998 lähtien. (Keith, s.a.) Uusimman *browscap.ini*-tiedoston voi ladata hänen projektisivuiltansa⁸, jossa voi myös kokeilla oman selaimen tunnistusta⁹.

Listaus 3.2: Esimerkki `get_browser()`-funktion käytöstä (W3Schools, s.a.-b)

```
1 <?php
2 $browser = get_browser( null , true );
3 print_r( $browser );
4 ?>
```

Palauttaa esimerkiksi:

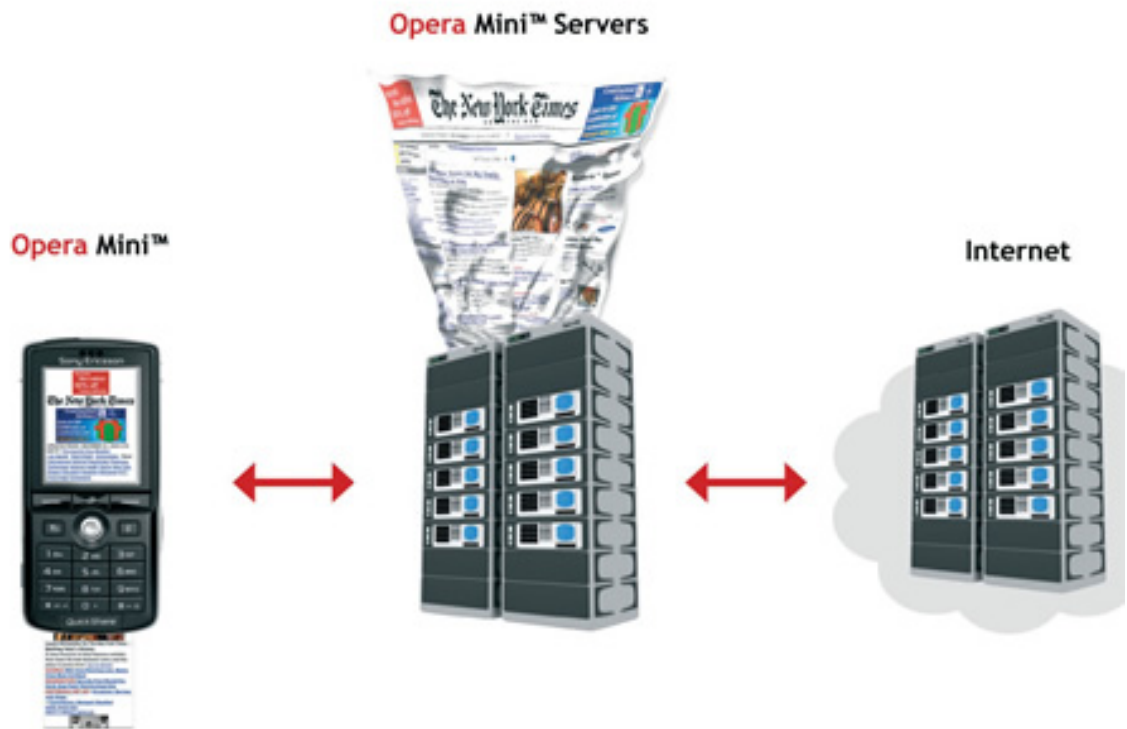
```
Array ( [browser_name_regex] => ^mozilla/.\.0 (compatible; msie 6\.0.*;
*windows nt 5\.1.*\.net clr.*).*$ [browser_name_pattern] =>
Mozilla/?0 (compatible; MSIE 6.0*;*Windows NT 5.1*.NET CLR*)*
[parent] => IE 6.0 [platform] => WinXP [netclr] => 1
[browser] => IE [version] => 6.0 [majorver] => 6 [minorver] => 0
[css] => 2 [frames] => 1 [iframes] => 1 [tables] => 1
[cookies] => 1 [backgroundsounds] => 1 [vbscript] => 1
[javascript] => 1 [javaapplets] => 1 [activexcontrols] => 1
[cdf] => 1 [aol] => [beta] => [win16] => [crawler] =>
[stripper] => [wap] => [ak] => [sk] => )
```

⁸<http://browsers.garykeith.com/downloads.asp>

⁹<http://browsers.garykeith.com/tools/your-browser.asp>

3.2.3 Opera Mini -selaimen tunnistus

Opera Mini¹⁰ on ilmainen mobiiliselain matkapuhelimille, ja sen suosio kasvaa jatkuvasti (Opera Software, 2009, s. 63-64). Vuoden 2010 alussa Opera Mini saavutti yli 50 miljoonan kuukausittaisen käyttäjän rajan. (Opera Software, 2010b) Suosiota lisää entisestään se, että Apple hyväksyi Opera Minin Applen suosittuun sovel-luskauppaan eli App Storeen, josta selaimen voi ladata Iphoneen ilmaiseksi (Digi-today, 2010). Laajan suosionsa johdosta myös Opera Minin käyttäjät on otettava huomioon ja siis tunnistettava UCL-hankkeen laboratorioympäristössä. Opera Mini käyttää *välityspalvelimia* matkapuhelimen ja Internetin välillä. Kuvassa 3.1 näky-vää toteutusta kutsutaan *client-server*-ratkaisuksi. Välityspalvelin optimoi verkko-palvelun sisällön Opera Miniä käyttävälle päätelaitteelle sopivaksi. Esimerkiksi kun välityspalvelin pakkaa websivuston sisällön ja skaalatessa kuvia pienemmäksi, vas-taanotettavan datan määrä pienenee ja täten selaaminen nopeutuu. (Eriksson, 2006) Nokia oman selaimen ja Operan selaimen välinen nopeusero voi olla yli 60% Operan hyväksi (Tietokone, 2010), puhelinmallista ja selainten versioista riippuen.



Kuva 3.1: Opera Minin välityspalvelimet prosessoivat sisällön uudelleen päätelaitteen ja webpalvelimen välillä (Bandeekar, 2005)

Opera Miniä käyttävän laitteen tunnistamisen tekee vaikeaksi se, että selain käyttää Opera Minin omia rekisteröimättömiä HTTP-otsikoita säännöllisten HTTP-

¹⁰<http://www.opera.com/mini/>

otsikoiden lisäksi. Opera Minin otsikoiden tarkoituksena on antaa mahdollisuus sivustojen ylläpitäjille kerätä tietoja käyttäjien päätelaitteista. Osa otsikoiden kentistä on pakollisia, jolloin ne lähetetään jokaisen HTTP-pyynnön yhteydessä. Erityisen hyödylliseksi `$_SERVER`-muuttuja osoittautuu juuri Opera Minin HTTP-otikoiden kanssa. `HTTP_USER_AGENT`-kentän sijaan voidaan Opera Minin tapauksessa käyttää `HTTP_X_OPERA_PHONE_UA`-kenttää, jonka käytöstä on esimerkki listauksessa 3.3. Muita hyödyllisiä kenttiä ovat `HTTP_X_OPERAMINI_PHONE`, joka palauttaa puhelimen valmistajan sekä mallin ja `HTTP_X_FORWARDED_FOR`, joka palauttaa pilkuilla eroteltuna kaikki tiedon siirrossa mukana olevat välityspalvelimet sekä tärkeimpänä Opera Miniä käyttävän laitteen IP-osoitteen. Näistä ainoastaan viimeinen on pakollinen kenttä, joten se lähetetään jokaisen HTTP-pyynnön yhteydessä. (Streng Hähre, 2007)

Lisää näitä niin sanottuja *X-kenttiä* on esitetty John Arnen blogikirjoituksesta (Arne, 2008). Kentistä suurin osa ei ole pakollisia, joten niiden toimivuus on laitekoh- taista ja tietoturvasyistä operaattorikohtaista, mikä havaittiin eri kenttiä testatessa. Listauksessa 3.3 on esimerkki rekisteröityjen HTTP-otsikoiden ja Opera Minin otsi- koiden eroista. Kenttä `HTTP_USER_AGENT` palauttaa User Agent -tiedot, jotka eivät kerro päätelaitteesta mitään. Samoin `REMOTE_ADDR` palauttaa satunnai- sen välityspalvelimen IP-osoitteen, eikä siis sen perusteella voi tehdä edes suuntaa- antavaa IP-paikannusta. Opera Minin otsikot puolestaan kertovat oikeat User Agent -tiedot, laitteen valmistajan ja mallin sekä IP-osoitteen.

Listaus 3.3: Esimerkki HTTP- ja Opera Minin HTTP-otsikoiden käytöstä

```

1 <?php // Rekisteröityjen HTTP-kenttien arvot
2 print('UA: '); echo $_SERVER['HTTP_USER_AGENT'];
3 print('<br/>');
4 print('IP: '); echo $_SERVER['REMOTE_ADDR'];
5 print('<br/><br/>');
6 //Opera Minin HTTP-kenttien arvot
7 print('Opera Minin UA: ');
8 echo $_SERVER['HTTP_X_OPERAMINI_PHONE_UA'];
9 print('<br/>');
10 print('Laite: ');
11 echo $_SERVER['HTTP_X_OPERAMINI_PHONE'];
12 print('<br/>');
13 print('Opera Mini -laitteen IP: ');
14 echo $_SERVER['HTTP_X_FORWARDED_FOR']; ?>
```

Nokia N73, jonka selaimena on Opera Mini 4.2, palauttaa esimerkiksi:

```

UA: Opera/9.80 (J2ME/MIDP; Opera Mini/4.2.14912/1186; U; en) Presto/2.2.0
IP: 94.246.126.225
```

Opera Minin UA: NokiaN73-1
Laitte: Nokia # N73
Opera Mini -laitteen IP: 80.186.31.241

Nokia N79, jonka selaimena on Opera Mini 4.2, palauttaa esimerkiksi:

UA: Opera/9.80 (J2ME/MIDP; Opera Mini/4.2.14912/1186; U; en) Presto/2.2.0
IP: 94.246.126.236
Opera Minin UA: Mozilla/5.0 (SymbianOS/9.3; U; Series60/3.2
NokiaN79-1/32.001; Profile/MIDP-2.1 Configuration/CLDC-1.1)
AppleWebKit/413 (KHTML, like Gecko) Safari/413
Laitte: Nokia # N79
Opera Mini -laitteen IP: 194.137.80.96

3.3 User Agent Profile

Työpöytäselainten ja matkapuhelimista Opera Minin tunnistukseen on siis olemassa valmiita ratkaisuja, mutta tavanomaisten matkapuhelinten ja niiden selainten tunnistamiseen eivät kyseiset menetelmät sovellu yhtä hyvin. Ongelmana on, että edellä mainituilla menetelmillä ei saada tarkempia tietoja päätelaitteesta. Tällaisia tietoja ovat esimerkiksi näytön resoluutio ja puhelimen tuki eri formaateille, kuten CSS-versioille, JavaScriptille ja kuville. Valmiit ratkaisut antavat kuitenkin keinon päästä lähemmäksi laitetietoja, sillä HTTP-otsikoilla voidaan PHP:n avulla saada linkki UAProf-tietoihin. UAProf eli User Agent Profile on standardi, joka sisältää tietoja päätelaitteesta, jota kutsutaan usein myös asiakasohjelman nimellä. Käytännössä tämän avulla saadaan siis tietoja selaimesta tai matkapuhelimesta ja sen ominaisuuksista. Kyseisen standardin on määritellyt Open Mobile Alliance¹¹ (entinen WAP Forum) erityisesti *Wireless Application Protocol* (WAP) -puhelimia varten. Se perustuu W3C:n *Composite Capabilities/Preferences Profile* (CC/PP) -määrittelyyn, joka on tarkoitettu yhteiseksi pohjaksi laitetta ja käyttäjää kuvaileville määrittelyille. CC/PP määrittelee laite- ja käyttäjätiedoille yleisen kehyksen näiden kuvailutietojen esittämiseen. Kehys koostuu *komponenteista* (engl. component), joilla on *attribuutteja* (engl. attributes). CC/PP ei kuitenkaan määrittele sanastoa laitteiden kuvailemiseen, vaan sen tekee UAProf. UAProf määrittelee sanaston lisäksi päättelysääntöjä erilaisten profiilien esittämiseen ja HTTP-protokollaan perustuvan protokollan kuvailutiedon välittämiseen. Listauksessa 3.4 on esimerkki UAProf-standardin määrittelemästä komponentista *laitteiston* (Hardware platform) ominaisuuksien kuvailemiseen. Muita UAProfin määrittelemiä komponentteja ovat esimerkiksi *sovellus* (Software platform), *selain* (Browser User Agent), *verkkoyhteys* (Network characteristics), *WAP-protokolla* (WAP characteristics) ja *Push*¹² (Push

¹¹<http://www.openmobilealliance.org>

¹²http://en.wikipedia.org/wiki/Push_technology [viitattu: 25.4.2010]

characteristics). Laittevalmistajat voivat myös määrittellä omia komponentteja tarpeidensa mukaisesti. (vrt. Wireless Application Protocol Forum, 2001, s. 19-27; Klyne et al., 2004; Huhtamäki, 2005, s. 53-54)

Listaus 3.4: Ote Nokia 6230i:n laitteistotiedoista (Developer's Home, s.a.)

```

1  ...
2  <prf:component>
3    <rdf:Description rdf:ID="HardwarePlatform">
4    ...
5      <prf:ScreenSize >208x208</prf:ScreenSize>
6    ...
7    </rdf:Description >
8  </prf:component>
9  ...

```

UAProf-tietoja säilytetään palvelimella, jota kutsutaan *profilien tallennuspaikaksi* tai *profilitietokannaksi* (engl. Profile Repository) (Wireless Application Protocol Forum, 2001, s. 14). Usein näitä tallennuspaikkoja ylläpitävät mobiililaittevalmistajat, kuten Nokia, mikä takaa tietojen saatavuuden ja luotettavuuden. UAProf-standardin mukaisten laitetietojen URL saadaan mobiililaitteista HTTP-otsikoista, joita päätelaitteet ja palvelimet välittävät tiedonsiirrossa antaen tietoja itsestään. Tiedonsiirron yhteydessä tämä URL saadaan usein poimittua \$_SERVER-muuttujalla *X-WAP-PROFILE-* tai *PROFILE-*kentän avulla. Kentän palauttama osoite viittaa suoraan laitteen UAProf-tietoihin. Esimerkiksi Nokia 6230i:n tapauksessa ensiksi mainittu *X-WAP-PROFILE-*kenttä palauttaa URL:n RDF-dokumenttiin¹³, joka sisältää päätelaitetietoja eri komponenttien ja edelleen attribuuttien sisällä. Esimerkiksi Nokia 6230i:n näytön koko nähdään edellä esitetystä listauksesta 3.4. (Developer's Home, s.a.) Nokia 6230i:n tapauksessa operaattorin määrittelemä oma laitetietoja kuvaileva komponentti on esimerkiksi *MMS* (Mms characteristics), joka kuvailee muun muassa, mitä audio- ja kuvaformaatteja puhelin tukee MMS-viestien osalta.

Laitetietojen verkko-osoitteeseen ja sieltä löytyvän RDF-dokumentin tietoihin voidaan useimmiten luottaa. Vaikka laitetiedot sijaitsevat kymmenissä eri sijainneissa laitevalmistajien määrän mukaisesti, laitetiedot saadaan automaattisesti poimittua UAProfin määrittelemän sanaston ja RDF-jäsentimeen avulla. Tämä on tosin työlästä ja vaikka UAProf-tietojen URL:n poimimiseen on olemassa useampia mahdollisia HTTP-kenttiä (vrt. Developer's Home, s.a.), ei ole varmuutta siitä, että jokaisesta laitteesta saataisiin kyseinen osoite. Kaikki laitteet eivät nimittäin tue UAProfia. (Abbott & Wilberg, s.a., s. 4; Developer's Home, s.a.) Tietojen sijaitse-

¹³<http://nds1.nds.nokia.com/uaprof/N6230ir200.xml>

minen laitevalmistajien sivuilla ei kuitenkaan täydellisesti takaa tietojen luotettavuutta, sillä on tapauksia, joissa vanha profiili on liitetty uuden puhelimen profiiliin sellaisenaan. Profiileissa voi esiintyä virheitä, eikä niitä ole helppo saada korjatuksi, sillä se edellyttäisi ottamaan yhteyttä suoraan laitevalmistajiin. (Passani, s.a.-b) Internetissä löytyy websivustoja¹⁴, joiden avulla voi selvittää oman selaimensa User Agentin eli päätelaitetiedon, mikä helpottaa esimerkiksi eri päätelaitteiden testauksessa.

3.4 Laitteistotiedonkeruukirjastot

Pelkän UAProf-linkin saaminen laitteesta ei siis ole tarpeeksi luotettava ja tehokas laitetietojen keräämiseen edellä esittyjen puutteiden vuoksi. Niinpä haluttiin selvittää, onko olemassa keskitettyjä tietokantoja, joihin on kerätty kaikki mahdolliset matkapuhelimet tietoineen. Saatavilla olevien UAProf-linkkien avulla voitiin varmistaa, että näiden laitteistotiedonkeruukirjastojen tiedot pitivät paikkaansa. Sopivien laitetietokantojen kartoituksessa haluttiin käsitellä mahdollisimman monta vaihtoehtoa, joista tarkoituksena oli valita hankkeeseen paras ja ilmainen vaihtoehto. Tällä kartoittamisella ja valmiin ratkaisun löytämisellä pyrittiin minimoimaan UCL-hankkeen toteutustyö.

3.4.1 DetectRight

Laitteistotietoa sisältäviä tietokantoja hakiessa ensimmäisenä vastaan tuli DetectRight¹⁵. Laitteita tietokannasta löytyy yli 10000. Webkehittäjille sivustolla on sekä PHP- että .NET-kirjastot laitetunnistusta varten. Lisäksi kumpaankin näistä on tehty käyttäjäystävälliset ohjeet, joissa neuvotaan kuinka nämä API:t saa käyttöönsä. (DetectRight.com, s.a.-a) Sivustolla voi hakea matkapuhelimia, ja sen avulla tarkistaa tietokannasta löytyviä laitteita ja nähdä joitakin niiden ominaisuuksia. Esimerkiksi Nokia N73:n tiedoista näkee laitteen kuvan ja ominaisuuksista muun muassa näytön korkeuden ja leveyden sekä tuen gif- ja jpeg-kuville. (DetectRight.com, s.a.-c)

DetectRight on osittain ilmainen. Se tarjoaa maksutta API:t ja rajattoman määrän tietokantahakuja, mutta esimerkiksi reaaliaikaisen datan hakeminen vaatii maksullisen Pro- tai Enterprise-version. Reaaliaikainen data mahdollistaa käyttäjän yksilöimisen, laitevalmistajan ja mallin tunnistuksen, geolokaation (maakohtainen tarkkuus) sekä laitetietojen profiloinnin HTTP-otsikoista. Lisäksi palvelusta maksamalla tulokset saa tallennettua joko XML- tai Excel-muodossa ja palveluun voi integroida omaa dataa. (DetectRight.com, s.a.-b)

¹⁴<http://whatsmyuseragent.com> ja <http://youruseragent.info/what-is-my-user-agent>

¹⁵<http://www.detectright.com>

Listauksessa 3.5 on esimerkki PHP API:n käytöstä. Käyttöön ottamiseen tarvitaan `detectRight.php`-tiedosto, joka tulee sijoittaa palvelimelle sellaiseen paikkaan, että se voidaan sisällyttää toteutukseen esimerkiksi `include_once()`-funktiolla. Tämä PHP-tiedosto sisältää valmiit funktiot, joiden avulla laitetiedot haetaan DetectRightin tietokannasta. PHP-tiedostoa ei tarvitse muokata muuten kuin lisäämällä siihen rekisteröitymisvaiheessa saatu lisenssiavain.

Listaus 3.5: DetectRightin PHP-rajapinnan käyttö

```

1 <?php //PHP API sisältää vain detectRight.php-tiedoston.
2 include_once("detectRight.php");
3 $dr = new DetectRight ();
4 $cp = $dr -> getCustomer($_SERVER);
5 //Tulostetaan kaikki löydetyt ominaisuudet omille riveilleen.
6 foreach ($cp as $key => $value) {
7     echo "$key : $value <br/>";
8 }
9 //Funktio laitteen näytön ominaisuuksien hakuun.
10 function getscreensizes($profile=array()) {
11     $this->clearError();
12     $sizes=array();
13     if (!$profile) {
14         if (!$this->properties) {
15             if ($this->headers) {
16                 $profile=$this->getProfileFromHeaders($this->headers);
17             } else {
18                 return false;
19             }
20         } else {
21             $profile=$this->properties;
22         }
23     }
24     $sizes['x']=DR_gv($profile, 'screenx', -1);
25     $sizes['y']=DR_gv($profile, 'screeny', -1);
26     $sizes['xchar']=DR_gv($profile, 'screencolumns', -1);
27     $sizes['ychar']=DR_gv($profile, 'screenrows', -1);
28     return $sizes;
29 } ?>

```

Yllä oleva koodi palauttaa päätelaitteesta näytönkokoan liittyvät tiedot, esimerkiksi Nokia N73:

```
screenx : 240, screeny : 320, screenrows : 6, screencolumns : 21
```

3.4.2 Handset Detection

Handset Detection¹⁶ on vastaavanlainen palvelu kuin DetectRight. Laitteita tietokannassa on yli 7000 ja valmistajia yli 200. Käyttöönnoton helpottamiseksi sivustolta löytyy PHP- ja .NET-kirjastojen lisäksi Python- ja Java-kirjastot. Lisäksi Handset Detection paikallistaa IP:n perusteella sijaintitiedot, kuten teleoperaattorin, maan, kaupungin, maakunnan sekä leveys- ja pituuspiirin. Sivustolla DetectRightin tapaan mahdollista selata lukuisia eri puhelimia ja niiden laitetietoja. Erona tässä DetectRightiin on, että kaikki tietokannassa oleva tieto on nähtävissä sivustolla eri kategorioihin jaoteltuina. (Handset Detection, s.a.-c) Lisäksi sivustolle rekisteröitymällä näkee omia tilastojansa päätelaitteiden tunnistuksista, esimerkiksi tunnistusten määrän, mobiililaitteiden osuuden, käytetyimmät selaimet ja päätelaitteet sekä tunnistettujen laitteiden sijainnit. Mahdollista on myös tehdä omia laskelmia ja graafeja lataamalla data itselleen CSV-muodossa joko sisältäen kaiken kerätyn datan tai vain mobiilidatan. (Handset Detection, s.a.-a)

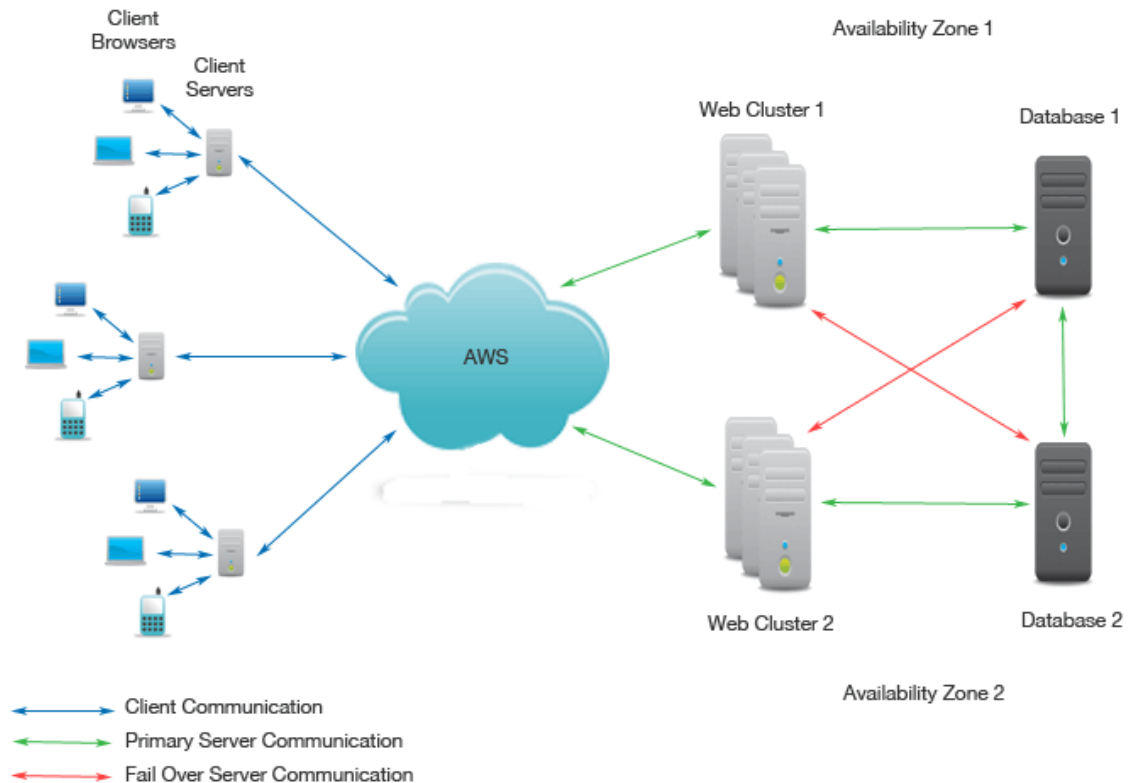
Handset Detection on maksullisuuden kannalta edullisempi vaihtoehto kuin DetectRight. Suurimman osan ominaisuuksista saa käyttöönsä maksutta, tietyin rajoituksin. Esimerkiksi laitetunnistuksen määrä on rajoitettu kuukausitasolla 10000:een, mikä tosin riittää UCL-hankkeen tarpeisiin. Dataa sen sijaan säilytetään vain kolmen kuukauden ajan, mutta ongelma on ratkaistavissa lataamalla data säännöllisin väliajoin talteen. Halvimmalla maksullisella lisenssillä eli Supporter-lisenssillä laitetunnistusten määrän saa nostettua 30000:een ja tätä kalliimmalla Blogger-lisenssillä 150000:een. Vastaavasti datan säilytys pitenee halvimmalla maksulla 12 kuukauteen ja tätä kalliimmalla lisenssillä data säilyy tietokannassa rajoittamattoman kauan. Lisäksi sivusto tarjoaa Pro-, Business- ja Enterprise-lisenssejä, joiden tuntuvin parannus on laitetunnistusten määrä kuukaudessa. (Handset Detection, s.a.-d)

Kuvassa 3.2 on Handset Detectionin oma näkemys palvelunsa toiminnasta eli siitä, miten laitetunnistus toimii. Prosessi sisältää neljä vaihetta (Handset Detection, s.a.-b):

1. Käyttäjä lähettää pyynnön websivustolle.
2. Webpalvelin lähettää pyynnön Handset Detectionin palvelimelle kysyen päätelaitteen tietoja. Tämän pyynnön aikana lasketaan verkon kuormitus siten, että pyyntö lähetetään parhaiten optimoidulle Handset Detectionin palvelimelle.
3. Handset Detection analysoi pyynnön ja vastaa siihen esimerkiksi, että kyseessä on Nokia N95 T-Mobilelta Saksasta ja palauttaa siihen liittyvät päätelaitetiedot.

¹⁶<http://www.handsetdetection.com>

4. Websivusto puolestaan päättää saamansa tiedon perusteella esimerkiksi:
- Uudelleenohjata käyttäjän mobiilisivustolle.
 - Lähettää mobiilisisältöä kuten pelejä, taustakuvia, soittoääniä.
 - Muokata sisältöä mobiililaitetta varten (kuten vaihtaa yksinkertaisempaan ulkoasuun ja luoda suorat linkit puheluita ja tekstiviestejä varten).
 - Yksinkertaisesti olla tekemättä mitään, jos vastaus on pätevä tai jos tarkoituksena on vain kerätä käyttäjän laitetiedot talteen.



Kuva 3.2: Handset Detectionin toimintaperiaate (Handset Detection, s.a.-b)

Handset Detectionin PHP API sisältää yhdeksän PHP-tiedostoa ja yhden config-tiedoston sekä ohjeet API:n käyttöön. Listauksessa 3.6 on osa `device.detect.php`-tiedoston sisältöä. Alussa kutsutaan `hdbase.php`-tiedostoa, joka sisältää varsinaisen tunnistuksen tekevät funktiot. Tämän jälkeen syötetään sivustolle rekisteröidyttyään saadut käyttäjätiedot. Tässä esimerkissä käytetään demo-tunnusta.

Listaus 3.6: Handset Detectionin PHP-rajapinnan käyttö

```

1 <?php
2 require_once( 'hdbase.php' );
3 $username=demo@handsetdetection.com
4 $secret=demo

```

```
5 //Haetaan laitteen tiedot.
6 $hd = new HandsetDetection();
7 $hd->detectInit();
8 $data = $hd->getDetect();
9 //Tulostetaan taulukkoon tallennetut muuttujat.
10 print_r($data);
11
12 //Tässä ote hdbase.php-tiedoston sisällöstä.
13 //Esimerkkinä funktio, joka hakee näytön leveyden ja korkeuden.
14 function getDeviceResolution() {
15     if (! isset($this->detectReply['display']['resolution_width']))
16     || ! isset($this->detectReply['display']['resolution_height']))
17     return FALSE;
18     $tmp = array();
19     $tmp['width']=
20     $this->detectReply['display']['resolution_width'];
21     $tmp['height']=
22     $this->detectReply['display']['resolution_height'];
23     return $tmp;
24 } ?>
```

Yllä oleva koodi palauttaa päätelaitteen näytön leveyden ja korkeuden, esimerkiksi Nokia N73:

```
Array(
  [resolution_width] => 240
  [resolution_height] => 320
)
```

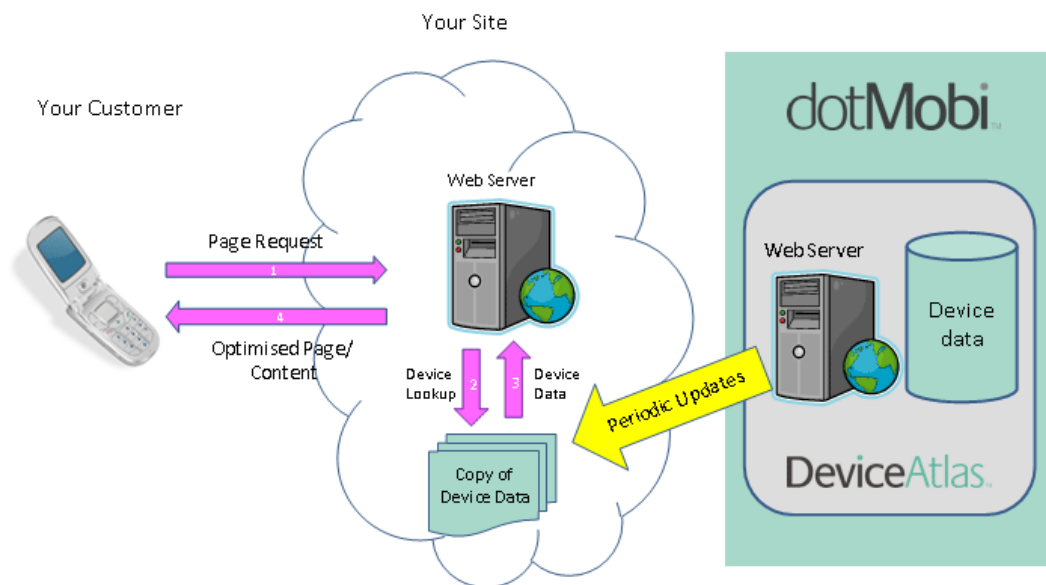
3.4.3 DeviceAtlas

DeviceAtlas¹⁷ tarjoaa edellisten tapaan suuren laitetietokannan, joka sisältää yli 5000 mobiililaitetta. Palvelun tukijoita ovat muun muassa Nokian Sony Ericsson ja Vodafone. Laitteiden tunnistusta varten sivustolla on viisi eri kirjastoa: Java, .NET, PHP, Python ja Ruby. Rekisteröityneille käyttäjille tarjotaan vuoden ilmaista lisenssiä, jolla voi tutkia dataa DeviceAtlaksessa omalla tilillään. Lisäksi tämä ilmainen Developer-lisenssi mahdollistaa valmiit tulosten analysoinnit ja minkä tahansa edellä mainitun API:n käytön. Sen sijaan tietokannan päivitystiheys ei ole säännöllinen, eikä tietokantaan voi tallentaa omaa dataa. Halvin maksullinen lisenssi on Basic-lisenssi, jolla saa edellisten ominaisuuksien lisäksi kuukausittaisen laitetietokannan päivityksen ja mahdollisuuden tallentaa palvelimelle omaa dataa.

¹⁷<http://deviceatlas.com>

Tätä kalliimmilla Standard- ja Premium-lisensseillä päivitystiheys paranee viikko- ja päivätasolle, mutta muita ominaisuuksia ei näillä lisensseillä saa. Hinnoittelemattomat Enterprise- ja OEM-lisenssit mahdollistavat muun muassa kaupallisen käytön, itsemääriteltävän päivitystiheyden, C++ API:n, MobiReady API:n laaduntarkkailuun sekä lokien analysoinnin. Näiden lisäksi OEM-lisenssi sallii jälleenmyynnin ja levityksen. (DeviceAtlas, s.a.-b)

Kuvassa 3.3 näkyy palvelun toimintaperiaate, joka on samankaltainen kuin edellisissä palveluissa. Matkapuhelin edustaa tässä webpalvelimen käyttäjää, joka lähettää webpalvelimelle pyynnön nähdä verkkosivun sisällön. Webpalvelin puolestaan katsoo matkapuhelimen tiedot päätelaitedatasta, joka päivitetään säännöllisen väliajoin DeviceAtlaksen palvelimelta. Tunnistuksen jälkeen webpalvelin optimoi sivuston ja sen sisällön matkapuhelimelle suunnitelluksi.



Kuva 3.3: DeviceAtlaksen toimintaperiaate (DeviceAtlas, s.a.-a)

3.4.4 Wireless Universal Resource File

Luca Passanin ja Andrea Trasattin kehittämä avoimeen lähdekoodiin perustuva Wireless Universal Resource File eli WURFL¹⁸ on *XML-dokumentti*, joka sisältää laite-tietoja yli 10000 mobiililaitteesta. WURFL:n XML-tiedostoon on tarkoituksena kerätä mahdollisimman paljon informaatiota kaikista mobiililaitteista, joilla on mahdollista käyttää WAP-sivustoja. WURFL:n avulla kerättävien tietojen toivotaan auttavan parempien sovellusten ja palveluiden kehittämiseen. WURFL:n käyttöön

¹⁸<http://wurfl.sourceforge.net>

on luotu erittäin monia API:ja ja työkaluja: PHP, Java, WALL (Passanin kehittämä Wireless Abstraction Library), WNG (Passanin kehittämä WALL Next Generation), Perl, .NET, Ruby, Python, C++ ja XSLT. Näistä kuitenkin enää vain neljää ensimmäistä päivitetään. (Passani, s.a.-c)

WURFL:n sisältämä informaatio on kerätty käyttäjiltä ympäri maailman. Kuka tahansa saa käyttää WURFL:ia mihin tahansa, myös kaupalliseen tarkoitukseen. Kuka tahansa voi myös päivittää laitetietoja noudattaen WURFL:n kehittäjien alkuperäisiä periaatteita. Kuten projektin sivuilla todetaan, tämän takia ei ole olemassa mitään varmuutta WURFL:n sisältämän datan oikeudellisuudesta. Toisaalta WURFL-projektiin päivittäin lähetetyt päätelaitetiedot kuitenkin varmistetaan UAProfin profiilin avulla ja tarvittaessa virheet korjataan. Esimerkiksi pyydetessä muutoksia Nokia N73:n versioon 5 (WURFL ID: nokia_n73_ver5), tietoja verrataan Nokian julkaisemiin N73:n UAProf-tietoihin¹⁹. (Passani, s.a.-c)

WURFL:n tarjoamien tietojen tarkasteluun ja päivittämiseen on tehty WURFL DB, jonka tarkoitus on nopeuttaa laitetietojen päivittämistä ja korjaamista ilman, että Passani tarkistaa ensin tiedot itse ja tallentaa ne tietokantaan manuaalisesti. Täyttämällä tietyt ehdot²⁰ käyttäjät voivat pyytää itselleen WURFL-kehittäjän asemaa ja näin saada käyttöliittymään tunnukset. Kehittäjillä on oikeus lisätä ja muokata päälaitetietoja sekä tämän jälkeen ladata itselleen WURFL:n XML-tiedosto. Peruskäyttäjää WURFL DB palvelee Mobile Device Informationin tapaan, tosin ilman päätelaittekuvia ja mahdollisuutta vertailla kätevästi laitteita keskenään. Käyttöliittymällä näkee kaikki WURFL:n sisältämät laitetiedot, mutta muita ominaisuuksia ei peruskäyttäjillä ole. Halutessaan uusia tai korjattavia laitetietoja voi lähettää joko suoraan sähköpostitse Passanille tai kirjoittamalla Yahoo-ryhmään nimeltä *WMLProgramming*²¹, jossa on myös WURFL:iin liittyviä alaryhmiä. (Passani, s.a.-a) Kehittäjiä varten on laadittu hyvin tarkat ohjeet²², jossa neuvotaan miten tietoja voi päivittää WURFL DB:n avulla. Tässä projektissa näihin ohjeisiin ei kuitenkaan ole ollut tarvetta perehtyä. Listauksessa 3.7 on WURFL:n XML-dokumentin skeema, josta näkyy miten laitetiedot esitetään WURFL:ssa. Listauksessa 3.8 on puolestaan ote Nokia N73:n WURFL-tiedoista, joista näkyy esimerkiksi näytön resoluutio. Listauksessa 3.9 on esimerkki WURFL:n PHP-rajapinnan käytöstä. Laitetiedoista haetaan PHP:n avulla näytön resoluution.

¹⁹<http://nds.nokia.com/uaprof/NN73-5r100.xml>

²⁰http://www.wurflpro.com/static/become_a_contributor.htm

²¹<http://tech.groups.yahoo.com/group/wmlprogramming>

²²<http://www.wurflpro.com/static/top.htm>

Listaus 3.7: WURFL:n DTD (Openwave Developer Network, s.a.)

```

1 <wurfl>
2   <devices>
3     <device ...>
4       <group ...>
5         <capability name="..." value="...">
6           :
7         </group>
8           :
9         </device>
10        :
11   </devices>
12 </wurfl>

```

Listaus 3.8: Nokia N73 WURFL:n XML-dokumentissa

```

1 <device id="nokia_n73_ver1" user_agent="NokiaN73"
2 fall_back="nokia_generic_series60_dp30"
3 actual_device_root="true">
4   <group id="product_info">
5     <capability name="pointing_method" value="joystick"/>
6     <capability name="uaprof"
7     value="http://nds.nokia.com/uaprof/NN73-1r100.xml"/>
8     <capability name="model_name" value="N73"/>
9     <capability name="device_os_version" value="9.1"/>
10    <capability name="uaprof2"
11    value="http://nds1.nds.nokia.com/uaprof/NN73r100.xml"/>
12  </group>
13  <group id="display">
14    <capability name="physical_screen_height" value="49"/>
15    <capability name="physical_screen_width" value="37"/>
16    <capability name="max_image_width" value="229"/>
17    <capability name="resolution_height" value="320"/>
18    <capability name="resolution_width" value="240"/>
19    <capability name="max_image_height" value="260"/>
20  </group>
21  <group ...>
22    :
23 </device>

```

Listaus 3.9: WURFL:n PHP-rajapinnan käyttö (Forum Nokia Wiki, 2009a)

```

1 <?php //Käyttöönotto: määritellään polut API:n tiedostoihin
2 require_once "WURFL_Installation/WURFL/WURFLManagerProvider.php";
3 $wurflConfigFile="WURFL_Installation/resources/wurfl-config.xml";
4 //Määritellään polku WURFL konfigurointitiedostoon.
5 $wurflManager =
6 WURFL_WURFLManagerProvider::getWURFLManager($wurflConfigFile);
7 $requestingDevice =
8 $wurflManager->getDeviceForHttpRequest($_SERVER);
9 //Haetaan halutut laitteen ominaisuudet, kuten resoluutio.
10 $width=$requestingDevice->getCapability("resolution_width");
11 $height=$requestingDevice->getCapability("resolution_height");
12 echo "width: " . $width;
13 print(', ');
14 echo "height: " . $height; ?>
```

Yllä oleva koodi tulostaa päätelaitteen näytön leveyden ja korkeuden, esimerkiksi Nokia N73:

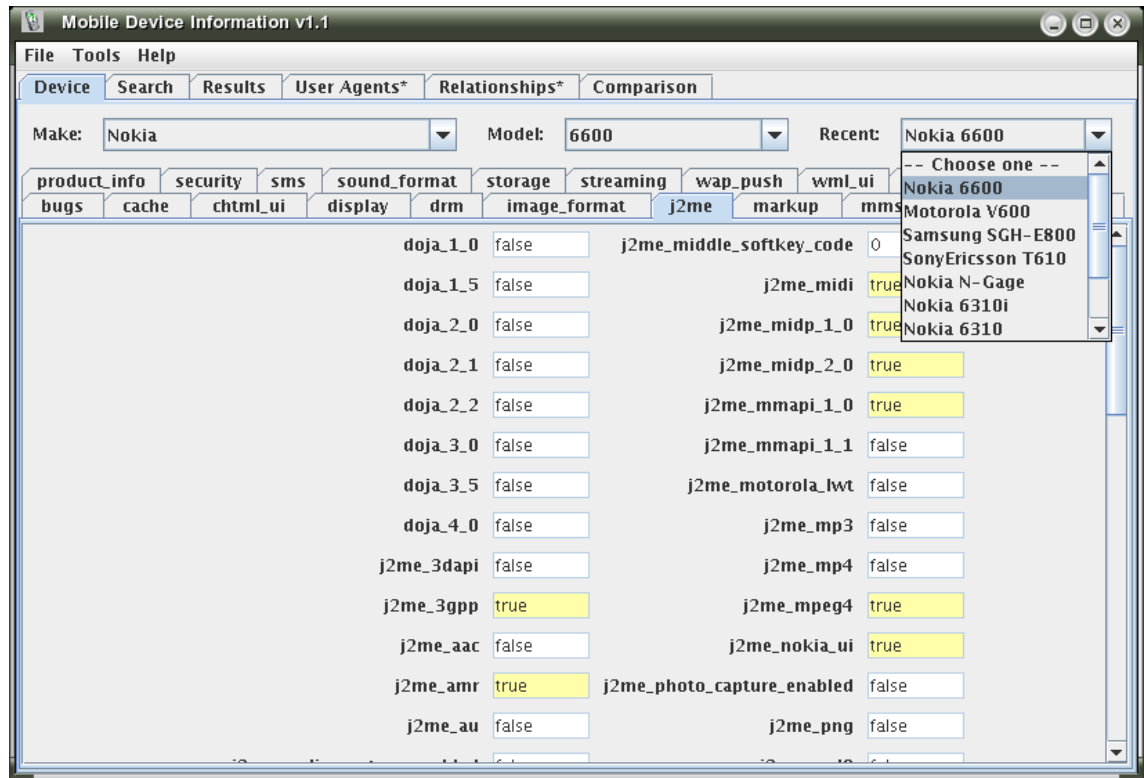
```
width: 240
height: 320
```

3.4.5 Mobile Device Information

Mobile Device Information eli mDevInf²³ ei varsinaisesti ole laitetietokirjasto, vaan se on Java-työkalu WURFL:n tarjoamien tietojen tarkasteluun. Sen avulla on helppo hakea esimerkiksi vain ne puhelimet, joissa on kosketusnäyttö tai tietty resoluutio. Tietokannasta löytyvät tulokset näkee kätevästi kategorioittain kyseisellä työkalulla ilman, että ne olisivat RDF- tai XML-muodossa. Ohjelman kehittäjä Jim McLachlan on Java-työkalun avulla halunnut tarjota nopean pääsyn WURFL-tietoihin. Tästä esimerkkinä mainitaan ylläpitäjän tarve selvittää, mitkä kaikki puhelimet tukevat hänen sivustonsa WAP-sisältöä. Ohjelman kerrotaan myös olevan kätevä työkalu tekniseen tukeen, kun halutaan selvittää matkapuhelinten ominaisuuksia. Lisäksi ohjelma soveltuu oman puhelimen tietojen tarkasteluun ja sen avulla voi vertailla eri puhelinten ominaisuuksia keskenään. (McLachlan, s.a.-a)

Kuvassa 3.4 on Mobile Device Informationin käyttöliittymä, jolla WURFL:n laitetietoja selataan. Device-välilehdellä voi selata kategorioittain eri ominaisuuksia. *Search-* ja *result-*välilehdet ovat nimensä mukaisesti laiteominaisuuksien hakuun ja tuloksiin. *User Agents-*välilehdellä voi hakea eri UAProf-tietoja. *Relationships* listaa puhelimet järjestykseen niiden käyttöliittymien mukaisesti. *Comparison-*välilehdellä voi puolestaan vertailla matkapuhelinten ominaisuuksia keskenään.

²³<http://mdevinf.sourceforge.net>



Kuva 3.4: mDevInf:n käyttöliittymä (McLachlan, s.a.-a)

Työkalua ei ole päivitetty sitten heinäkuun 2006, ja sivuston uusin uutinen on lokakuulta 2007 (McLachlan, s.a.-b). Työkalun käyttämä tietokanta on kuitenkin päivitettävissä lataamalla uusin WURFL:n XML-dokumentti.

3.5 Laitteistotiedonkeruumenetelmien vertailu

UCL-hanketta varten tutustuttiin siis tarkemmin viiteen eri laitteistotiedonkeruukirjastoon ja automaattiseen selaintunnistukseen. Eri vaihtoehtoja oli täten riittävästi, joten seuraavaksi piti karsia joukosta sopivin vaihtoehto hankkeen tarpeisiin. Taulukkoon 3.1 on koottu eri vaihtoehdot yhteenvedoksi eli poimittu jokaisesta oleellisimmat tiedot. Tärkein näistä tiedoista on lisenssi, sillä hankkeen ehtona on, että laitetietokirjaston käytön tulee olla maksutonta. Lisäksi oleellinen ominaisuus on eri vaihtoehdot käyttörajapinnaksi siten, että käyttöönotto on mahdollisimman helppoa. Laitetietojen määrä tietokannassa on myös hyvä tietää, jotta mahdollisimman moni laite tulisi tunnistetuksi.

Taulukko 3.1: Laitteistotiedonkeruukirjastojen tärkeimmät ominaisuudet

Nimi	Tietokannan koko	Kirjastot	Lisenssi
get_browser() ja \$_SERVER-muuttuja	Ei tiedossa	Ei varsinaista API:a, tuki PHP:lle ja .NET:lle	Ilmainen
User Agent Profile	Ei tiedossa	Ei varsinaista API:a, tiedot jäsennettävissä RDF-jäsentimellä	Ilmainen
DetectRight	Yli 10000	PHP, .NET	Osittain ilmainen
Handset Detection	Yli 7000	PHP, .NET, Python, Java	Ilmainen (yli 10000 tunnistusta/kk ja 3kk:ta pidempi datan säilytys maksullista)
DeviceAtlas	Yli 5000	PHP, .NET, Python, Java, Ruby	Ilmainen (säännöllinen päivitys maksullista)
WURFL	Yli 10000	PHP, .NET, Python, Java, Ruby, WALL, WNG, Perl, C++, XSLT	Open Source
Mobile Device Information	WURFL:n tietokanta	Ei API:a, pelkkä Java-työkalu	Open Source

Ensimmäisenä vaihtoehtoista on helppo karsia pois Mobile Device Information, sillä kyseinen Java-työkalu ei sovellu kuin WURFL:n tietojen tarkasteluun, eikä automaattinen laitetunnistus onnistu sillä ollenkaan. Listan ensimmäiset eli get_browser() ja \$_SERVER-muuttuja eivät puolestaan sellaisenaan riitä laitetunnistamiseen, sillä niiden avulla ei saada tarkempia tietoja, kuten näytön resoluutiota. User Agent Profile -sanaston kuvaamiin RDF-dokumentteihin päästään käsiksi UAProf-linkkien avulla. Dokumenteista tiedot saadaan jäsennettyä RDF-jäsentimellä, mutta tämä ei onnistu kaikista laitteista luvussa 3.3 esitetyin perustein. DetectRightin etuja ovat suuri tietokanta, helppo käyttöönotto, ohjeet ja monipuoliset laitetiedot. Sen sijaan palvelun ominaisuudet ilmaisella lisenssillä ovat varsin heikot, sillä esimerkiksi reaaliaikaisen datan käyttöön vaaditaan maksullinen lisenssi. Lisäksi DetectRightin lisenssien hinnat ovat muiden palveluiden maksullisiin lisensseihin verrattuna huomattavasti kalliimmat. Handset Detection tarjoaa sekin riittävän kokoisin tietokannan laitetunnistamiseen. Lisäksi PHP API julkaistiin laitetietokantojen kartoitusvaiheessa ja sen käyttöönotto sujui vaivattomasti. Samoihin aikoihin julkaistiin myös palvelun lisenssien hinnat, mikä kertoi osaltaan siitä, ettei palvelu ole täysin ilmainen. WURFL:n tietokanta on DetectRightin tapaan erittäin kattava ja erilaisia käyttörajapintoja on useita. Lisäksi palvelu perustuu avoimeen

lähdekoodiin, ja tietokanta päivittyy jatkuvasti. Kartoituvaiheen lopussa löytynyt DeviceAtlas vaikutti myös hyvältä vaihtoehdolta, mutta sen testaamiseen ei ollut tarpeeksi aikaa.

Näiden tietojen ja testauskokemusten perusteella kaksi vartenotettavaa vaihtoehtoa olivat WURFL ja DetectRight. WURFL valittiin näistä UCL-hankkeeseen, sillä se on täysin ilmainen ja laitetietokanta on suuri. Lisäksi merkittävänä perusteluna valinnalle oli Drupal-moduuli Mobile Plugin, joka käyttää WURFL-tietoja. Drupal²⁴ on PHP-pohjainen avoimen lähdekoodin sisällönhallintajärjestelmä ja sovelluskehitysalusta. Drupal koostuu monista eri toiminnoista, joten se on jaettu osiin, joita kutsutaan moduuleiksi. Moduuleja ovat esimerkiksi kommentointi, käyttäjäprofiili ja foorumit. Drupalissa voi kytkeä pois ja päälle eri moduuleja sekä asentaa muiden kehittämiä moduuleja. (Sani, 2008) Lisää Drupalista on luvussa 5.1. Moduulin valinnassa huomioitiin se, ettei WURFL sovellu työpöytäkoneiden selainten tunnistamiseen yhtä hyvin kuin DetectRight. Kumpikaan näistä ei puolestaan tunnista Opera Miniä käyttäviä päätelaitteita. Niinpä BrowserCapsin ja \$_SERVER-muuttujan käyttö otettiin WURFL:n rinnalle, jotta tiedot saataisiin mahdollisimman monesta laitteesta talteen.

3.6 Laitetunnistuksen toteuttaminen Drupalissa

UCL-projektin alussa ei ollut selvillä, mille alustalle verkkoympäristö toteutetaan. Yksi ratkaisevista tekijöistä oli sopivan laitteistotiedonkeruukirjaston valitseminen. WURFL:n valitseminen ja sopivien Drupal-moduulien löytäminen helpottivat valintaa, josta on tarkemmin luvussa 5.1. UCL:n laitetunnistuksen toteutukseen valittiin Mobile Plugin- ja Browscap-moduuli.

3.6.1 Drupal Mobile Plugin

Drupal-moduuleista valittiin laitetunnistukseen Mobile Plugin²⁵, joka valittiin siksi, että hankkeen laboratorioympäristö toteutetaan Drupaliin. Moduulin käyttöönotto valmiiseen ympäristöön sujuu täten vaivattomasti. Moduuli käyttää WURFL:n tarjoamia tietoja, joten siltäkin osin moduuli oli helppo valita käyttöön. Lisäksi moduulin ominaisuudet sopivat hankkeen vaatimuksiin. Tällaisia ovat automaattinen päätelaitteen tunnistus, mahdollisuus vaihtaa näkymää työpöytä- ja mobiiliversioiden välillä sekä luoda eri päätelaitteille erilaisia profiileja. Profilien tärkein toiminto on mahdollisuus muuttaa CSS- ja JS-tiedostojen käyttöä päätelaitteen ominaisuuksien, kuten selaimen mukaan. Myös moduulin julkaisupäivä (26.5.2009) viestitti sen

²⁴<http://drupal.org>

²⁵<http://drupal.org/project/mobileplugin>

ajankohtaisuudesta ja tulevia päivityksiä pidettiin mahdollisena. Tämän lisäksi moduulin kehittäjä oli tehnyt Mobile Pluginia varten selkeän asennusohjeen ja optimoidun mobiiliteeman eli mobiililaitteille tarkoitetun sivuston ulkoasun nimeltään Mobile Garland²⁶, mitkä helpottivat huomattavasti käyttöönottoa ja moduulin toimivuuden arviointia hankkeen tarpeisiin.

Vertailemalla liitteen 1 esimerkkejä 1 ja 2 huomataan, että Mobile Plugin ei WURFL:n tietokannan avulla saa läheskään yhtä hyviä tuloksia työpöytäselaimesta kuin matkapuhelimesta. Firefox ja sen versio näkyvät tuloksissa, mutta ne pitäisi vielä erotella *HTTP_USER_AGENT*-kentästä. Lisäksi tuloksissa näkyy vääriä tietoja, kuten että kyseessä on kosketusnäyttö, jonka resoluutio on 770x300. Nokia N73:n kohdalla tiedot täsmäävät, ja tietokannasta löytyy kaksi eri linkkiä Nokian tarjoamiin UAProf-tietoihin. Myöskään Opera Miniä käyttävän päätelaitetietoja ei saada moduulilla selville. Liitteen 1 esimerkistä 3 nähdään tämän lisäksi, että selain tunnistuu oikein, mutta resoluutioksi ilmoitetaan 128x92. Sama resoluutio saadaan tulokseksi päätelaitteesta riippumatta.

3.6.2 Drupal Browscap

Funktion `get_browser()` käyttöönoton helpottamiseksi on tehty Drupal-moduuli nimeltään Browscap²⁷. Moduulin ansiosta browscap.ini-tiedoston sisältö tallennetaan suoraan palvelimen tietokantaan, mikä vähentää konfiguroinnin tarvetta. Lisäksi ylläpitäjän ei tarvitse päivittää käsin browscap.ini-tiedostoa, vaan moduuli tekee päivityksen automaattisesti esimerkiksi viikoittain. Lisäksi ylläpitäjä voi kytkeä päälle käytönseurannan eli tilastoida sivustolla vierailleiden käyttäjien selaintiedot. Ainut ehto moduulin käytölle on, että palvelimen täytyy pystyä tekemään HTTP-pyyntöjä ulospäin, jotta browscap.ini-tiedoston voi ladata.

Liitteen 2 esimerkissä 1 on tulokset `get_browser()`-funktion käytöstä Browscap-moduulissa. Esimerkkilaitteena on Nokia N73, jonka tietoja saadaan moduulilla varsin heikosti selville. Käytännössä ainoat hyödylliset tiedot ovat selain ja tieto siitä, onko kyseessä mobiililaitte. Sen sijaan selaintunnistuksessa moduuli toimii puolestaan hyvin. Liitteen 2 esimerkistä 2 nähdään, että selaimen nimi ja versio sijoitetaan kätevästi omiin muuttujiinsa. Lisäksi moduuli tunnistaa muun muassa käyttöjärjestelmän ja sen, onko kyseessä langaton laite. Liitteen 2 esimerkin 3 mukaisesti myös Opera Mini tunnistetaan oikein, mutta päätelaitteesta ei saada selville edes käyttöjärjestelmää.

²⁶http://drupal.org/project/mobile_garland

²⁷<http://drupal.org/project/browscap>

4. PAIKANNUS

Kuten luvussa 2 kerrotaan, laitetietojen lisäksi käyttökokemuksia kerätessä myös paikkatieto halutaan saada talteen. Paikkatietoa voidaan hyödyntää monilla eri aloilla, kuten teollisuudessa, logistiikassa ja kuljetuksessa, mediassa, turvasektorilla (esim. poliisi) sekä kaupoissa. UCL:n näkökulmasta paikkatiedon avulla voidaan tutkia, mitä palveluja käytetään eri paikoissa eli nähdään, miten käyttöpaikka vaikuttaa eri palveluiden käyttöön. Esimerkiksi työpaikan ja kodin välillä on varmasti eroja verkkopalveluiden käytössä. Lisäksi käyttöpaikalla voi olla negatiivinen tai positiivinen vaikutus käyttökokemukseen. Esimerkiksi kotona ilman kiireitä on todennäköisesti helpompi käyttää pankkipalveluita verkossa kuin käyttää niitä matkapuhelimella kaupungin keskustan ruuhkassa. Myös käyttökokemusten hajaantuminen maantieteellisesti on tutkijan näkökulmasta kiinnostavaa. Missä palveluita käytetään ja kuinka tarkasti käyttäjät ovat valmiita kertomaan oman sijaintinsa?

4.1 Paikannuksen hyödyt ja ongelmakohdat

Tässä työssä *paikkatieto* tarkoittaa Innasen & Saarimäen (2009, s. 332) määritelmän mukaisesti tietoa, joka ilmaisee liittymän tai päätelaitteet maantieteellisen sijainnin ja jota käytetään muuhun tarkoitukseen kuin verkkopalvelun tai viestintäpalvelun toteuttamiseen. Paikkatiedoilla voidaan ilmaista liittymän tai päätelaitteen leveysaste, pituusaste ja korkeus, matkan suunta, paikkatiedon tarkkuus, se osa verkkoa, jossa liittymä tai päätelaite paikannetaan tietyllä hetkellä, ja paikkatiedon tallentamisen ajankohta. *Paikannettavalla* henkilöllä puolestaan tarkoitetaan luonnollista henkilöä, jolla on hallinnassaan paikannettava liittymä tai päätelaite.

UCL-verkkoympäristössä paikkatiedon keräämiselle on muutamia perusteluita. Paikkatiedon avulla on esimerkiksi helpompi myöhemmin palata muokkaamaan tekemäänsä käyttökokemusta, sillä käyttäjän ei tarvitse muistella käyttöpaikkaa. Käyttäjä ei välttämättä ole täysin varma, missä hän palvelua käyttää. Tähän auttaa paikkatiedon tallentaminen lomakkeen täyttövaiheessa. Lisäksi vaikei paikannus olisi välttämättä kovinkaan tarkka, se helpottaa käyttäjää, sillä karttanäkymässä on helpompi siirtää osoitinta läheltä todellista sijainti kuin aloittaessa maapallonäkymästä. Tällöin ei tarvitse tarkentaa karttaa lukuisia kertoja, mikä voi olla hyvin työlästä mobiililaitteella. Samoin, jos tieto saadaan automaattisesti lomakkeelle, optimtapauksessa käyttäjän ei tarvitse kuin varmistaa sijainti karttanäkymästä. Käyt-

täjän ei siis tarvitse syöttää koordinaatteja tai katuosoitetta lomakkeelle. Paikkatiedon tallentaminen käyttökokemuksen yhteyteen voi motivoida käyttäjiä käyttämään UCL:ää, kuten on tapahtunut useissa paikkatietoa hyödyntävissä palveluissa, joista lisää luvussa 4.3.

Paikkatietoon liittyy myös ongelmakohtia, joista palvelun kehittäjän näkökulmasta hankalin on paikkatiedon arkaluonteisuus. Etenkin automatisoidussa keräämisessä paikkatiedon saaminen kaikista mahdollisista laitteista on työlästä, ellei jopa mahdotonta. Laitteita on valtava määrä, eikä läheskään kaikissa ole ominaisuuksia, joiden avulla paikannuksessa päästäisiin edes kymmenien metrien tarkkuuteen. GPS-vastaanotin on yleistynyt matkapuhelimeissa, mutta maaliskuun alussa 2010 esimerkiksi Nokian 90 uusimmasta puhelimesta¹ vain 35:ssä oli GPS-paikannus. WLAN oli 30:ssä ja nämä molemmat olivat 23 puhelimesta. Kun mukaan otetaan vanhemmat puhelinmallit, joita on vielä paljon käytössä, GPS:n prosentuaalinen osuus vähenee huomattavasti. Lisäksi muidenkaan valmistajien tilanne ei näytä olevan merkittävästi parempi. Samaan aikaan tarkistetuissa Sony Ericssonin malleissa² GPS-vastaanottimien määrän suhde kaikkiin uusimpiin puhelinmalleihin oli 16/62 ja Samsungilla³ suhdeluku oli 10/38. Paikannuksessa on siis otettava huomioon, ettei läheskään kaikilla ole GPS-vastaanotinta puhelimesta ja vaikka olisikin, ei voida olettaa, että sitä pidettäisiin koko ajan päällä. Niinpä paikannuksen toteutusta varten on kartoitettava useita vaihtoehtoja, jotta paikkatieto saataisiin mahdollisimman monesta päätelaitteesta talteen.

Paikkatiedon tallentamiseen liittyy palvelun ylläpitäjän näkökulmasta myös vastuukysymyksiä, sillä tietosuojalaki estää paikantamisen ilman käyttäjän suostumusta: "Se, joka on ottanut vastaan tai muutoin saanut tiedon paikkatiedosta, jota ei ole hänelle tarkoitettu, ei saa ilman paikannettavan suostumusta ilmaista tai käyttää hyväksi paikkatietoa tai tietoa sen olemassaolosta" (Innanen & Saarimäki, 2009, s. 32). Niinpä UCL-verkkoympäristössä ei käyttäjille saa jäädä epäselväksi, että laitetietojen lisäksi myös paikkatieto kerätään talteen. Paikannus voidaan suorittaa ilman paikannettavan suostumusta vain hätätapauksissa poliisin tai pelastusviranomaisen määräyksestä (Innanen & Saarimäki, 2009, s. 332), joten UCL:ssä paikannus tehdään vain käyttäjien suostuttua siihen. Lisäksi käyttäjillä on mahdollisuus poistaa jälkepäin sijaintinsa verkkoympäristöstä.

¹<http://www.nokia.fi/tuotteet/kaikki-puhelimet>

²<http://www.sonyericsson.com/cws/products/mobilephones?cc=fi&lc=fi>

³<http://www.samsung.com/fi/consumer/mobile/mobilephones/mobilephones/index.idx?pagetype=subtype>

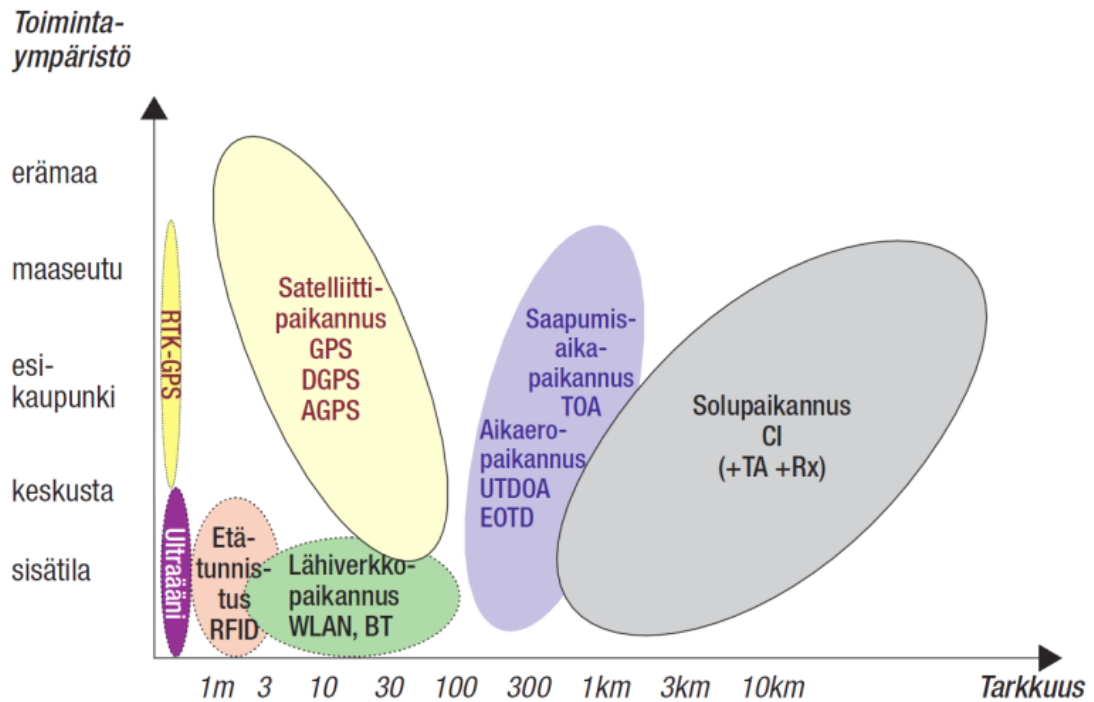
4.2 Paikannusvaihtoehtoja

Puhelun soittaminen tai muu viestintä eivät aina ole edellytyksiä paikantamiselle. Paikantaminen voi perustua esimerkiksi tukiasemapaikannukseen, satelliittipaikannukseen, langattomiin lähiverkkoihin, paikannusrannekkeisiin, RFID-tunnisteisiin, bluetooth-yhteyteen tai muihin paikannuslaitteisiin. (Innanen & Saarimäki, 2009, s. 331) Tässä työssä käytetään Rainion (2003) mukaista paikannusmenetelmien jaottelua, jossa menetelmät voidaan jaotella paikannusta tukevan infrastruktuurin mukaisesti kolmeen pääryhmään: *verkko-, lähi- ja satelliittipaikannus*. Verkkopaikannuksen etuna on matkapuhelinten määrä, jolloin kysyntä verkkopaikannukseen perustuville mobiilipalveluille on myös suuri. Verkkopaikannuksen tarkkuudessa ei kuitenkaan päästä lähelle satelliittipaikannusta, jonka ongelmia puolestaan ovat laitteen virrankulutus, käynnistymisen hitaus ja paikannuksen toimivuus kaupunkiympäristöissä ja erityisesti sisätiloissa. Lyhyen kantaman radiotekniikoiden eli lähipaikannuksen menetelmät on puolestaan kehitelty sisätiloja tai niihin verrattavia kaupunkitiloja ajatellen, sillä niissä verkko- ja satelliittipaikannuksen tarjoama tarkkuus ei riitä. (Rainio, 2003, s. 4)

Paikannustarkkuuteen vaikuttaa ratkaisevasti se, kuinka tarkasti kyetään mittaamaan aikaa. Paikannuksen tarkkuudella on puolestaan merkitystä sovelluksille, joita paikannuksen avulla toteutetaan. Esimerkiksi verkkopaikannus riittää moniin tiedonhakutarpeisiin ja käyttäjien paikantamiseen. Sen tarkkuus ei kuitenkaan riitä opastettaessa käyttäjä paikasta paikkaan, mihin vaaditaan vähintään satelliittipaikannuksen tarkkuutta. Satelliittipaikannuksen tarkkuuden voidaan sanoa riittävän kertomaan, missä kadunkulmassa käyttäjä on ja solupaikannuksen (yksi verkkopaikannuksen menetelmistä), missä kaupunginosassa. Kuvassa 4.1 on esitelty eri paikannusmenetelmien toimintaympäristöt ja tarkkuudet. Verkkopaikannuksen tarkkuus vaihtelee merkittävästi riippuen käyttöympäristöstä. Kaupunkien keskustoissa tarkkuudessa päästään muutamaan sataan metriin ja taajamissa pysytään vielä noin kilometrin tarkkuudessa, mutta maaseudulla tulos vaihtelee parista kilometristä pariinkymmeneen kilometriin. Satelliittipaikannuksen tarkkuuden voidaan yleisesti sanoen olevan noin kymmenen metriä, mutta kaupunkien rakenteiden heikentävät vaikutukset voivat vaikuttaa tarkkuuteen kymmenillä metreillä. Lähipaikannuksen avulla sisätiloissa päästään muutaman metrin tarkkuuteen. Tätäkin tarkempiin tuloksiin on mahdollista päästä ultraäänipaikannuksella ja ammattimaisella satelliittipaikannuksella, joilla voidaan saavuttaa senttimetrin tarkkuus. (Rainio, 2003, s. 4-5)

4.3 Verkkopaikannus

Kun puhelimen on auki, matkapuhelinverkko ylläpitää paikannustietoa jatkuvasti tukiasemaryhmän tarkkuudella, ja puhelun aikana paikkatieto saadaan tukiaseman



Kuva 4.1: Paikannusmenetelmien toimintaympäristö ja tarkkuus (Rainio, 2003, s. 5)

eli solun tarkkuudella. Jos järjestelmään lisätään erillinen paikannuspalvelin, voidaan matkapuhelin paikantaa solun tarkkuudella kytkemättä puhelua. Tämä onnistuu esimerkiksi tekstiviestin lähetyksen tai GPRS-pakettien siirtämisen yhteydessä. On kuitenkin otettava huomioon, että puhelimen jatkuva seuraaminen kuluttaa merkittävästi verkon resursseja. (Rainio, 2003, s. 5)

Yleisimpiä verkkopaikannusmenetelmiä ovat (Rainio, 2003, s. 5):

- solupaikannus (CI, Cell Identification)
- solupaikannus täydennetyllä ajastusennakko- ja signaalin voimakkuustiedoilla (CI+TA+Rx; TA, Timing Advance; Rx, signal level)
- saapumiskulmapaikannus (AOA, Angle of Arrival)
- saapumisaikapaikannus (TOA, Time of Arrival)
- kulkuajakaeropaikannus (EOTD, Enhanced Observed Time Difference)
- saapumisaikaeropaikannus (TDOA, Time Difference of Arrival)
- korrelaatiopaikannus (LF, Location Fingerprint)

Näistä tässä työssä käsitellään vain solupaikannus, joka on hyödyllinen varajärjestelmä silloin, kun GPS (tai A-GPS) ei kykene paikannukseen esimerkiksi sisätiloissa.

Solupaikannuksen avulla on mahdollista saada paikkatieto myös niistä matkapuhelimista, joissa ei GPS:ää ole. Näin saatu paikkatieto on huomattavasti epätarkempi kuin GPS:n avulla saatu, mutta kaikki käyttäjät eivät välttämättä ole valmiita pitämään GPS:ää päällä jatkuvasti. Syynä voi olla esimerkiksi yksityisyyden menettämisen pelko tai se, että GPS kuluttaa huomattavasti nopeammin matkapuhelin akkua sen ollessa päällä. Tämän ja toistaiseksi varsin hitaasti yleistyvän A-GPS:n vuoksi solupaikannus tulee olemaan operaattoreiden käytössä vielä kauan. (Paikannus.com, s.a.-b) GPS:n ja A-GPS:n toiminnasta on kerrottu tarkemmin luvussa 4.5.

Solupaikannus perustuu matkapuhelimen paikannukseen tukiaseman kuuluvuusalueen mukaan. Se on yksinkertaisin menetelmä paikannukseen, sillä solupaikannus ei vaadi merkittäviä muutoksia verkon laitteisiin. Tukiasemien paikat ovat kiinteitä, joten paikantamiseen riittää kartta tukiasemien tai niiden sektoreiden peittoalueista. (Paikannus.com, s.a.-b) Tukiasemien tarkat sijainnit ovat arkaluonteista tietoa, joten operaattorit pyrkivät pitämään ne omana tietonaan. Internetissä on useita sivustoja, joille on vapaaehtoisten voimin kerätty tukiasemasolujen koordinaatteja. Niinpä tukiaseman solunumeron selvittyä on mahdollista hakea tukiaseman koordinaatit tietokannasta. Solupaikannuskirjastoista on kerrottu tarkemmin luvussa 4.6.

Solupaikannuksen haittapuolena on sen epätarkkuus, sillä paikannuksen tarkkuus on suoraan verrannollinen tukiasemansolun kokoon. 2G-verkossa pienten ja keski suurten GSM-solujen peittoalue on noin 250-500 metriä ja suurten solujen peittoalue voi maaseudulla ulottua noin 30 kilometriin. Lisäksi paikannustarkkuutta heikentää se, ettei matkapuhelinta palveleva tukiasema ole välttämättä aina lähin mahdollinen. (Kummala, Kulmala, Hautala, Oinas, & Holm, 2001, s. 20; Pulli, Posti, & Tapaninen, 2009, s. 59) Toisin kuin 2G-verkossa, matkapuhelin voi olla 3G-verkossa samanaikaisesti yhteydessä useamman tukiaseman kanssa. Niinpä ei voida tietää tarkasti, minkä tukiaseman peittoalueella matkapuhelin on. Solupaikannuksen tarkkuutta 3G-verkossa voidaan kuitenkin parantaa tilastollisilla menetelmillä. Yhtenä varteenotettavana vaihtoehtona pidetään *aikaeromittausta* (OTDOA, Observed Time Difference of Arrival), jossa tukiaseman kuuluvuutta parannetaan toisia tukiasemia *hetkellisesti vaimentamalla* (IPDL, Idle Periodic Down Link). Tällä on mahdollista saavuttaa noin 50-100 metrin tarkkuus. (Paikannus.com, s.a.-b; Rainio, 2003, s. 7-8)

Solupaikannus on ollut UCL-projektin kannalta mielenkiintoinen aihe, sillä esimerkiksi hätäkeskuslaitos⁴ käyttää tarvittaessa solupaikannusta hädässä olijan paikannukseen. Hätäkeskuslaitoksen mukaan soittaja ei usein tiedä omaa olinpaikkaansa. Paikannus toimii hätäkeskusjärjestelmässä nappia painamalla ja se kestää noin 10 sekuntia. Hätäpaikannus tehdään GSM-liittymänumeron perusteella ja paikkatiedon järjestelmälle palauttaa liittymän kotioperaattorin paikannuspalvelin. Pai-

⁴<http://www.hatakeskus.fi/>

kannus epäonnistuu melko harvoin. Tällöin useimmiten yritetään paikantaa SIM-kortitonta tai suljettua liittymää. Solutieto saadaan puhelimesta vielä noin vuorokauden ajan puhelimen sulkemisen jälkeen. Myös SIM-kortiton puhelin on mahdollista tunnistaa, sillä hätäpuhelua soittaessa sen puhelinnumero on järjestelmässä jonkun operaattorin keskusnumero. Mittauksen tarkkaa tekniikkaa ei mainita, mutta kerrotaan että se tehdään "tukiasemien avulla signaalin vaimentuminen yms. huomioiden." Paikannustarkkuudet eri alueilla ovat samaa luokkaa kuin edellä mainitut Rainion (2003, s. 4) mukaiset lukemat. Kaupunkialueilla hätäpaikannuksen tarkkuus on noin 50-400 metriä, esikaupunkialueella noin 100-1000 metriä, taajamien ulkopuolella ja isoilla vesistöalueilla noin 1-5 kilometriä sekä harvaanasutuilla alueilla ja merialueilla jopa kymmeniä kilometrejä. (Hätäkeskuslaitos, s.a.)

Solupaikannuksen käytön perustelut ovat niin UCL-projektin kuin hätäkeskuksen kannalta samat, sillä GPS-ominaisuutta ei ole vielä monessa matkapuhelimesa. Niinpä hätäpaikannukseen ei ainakaan vielä ole kehitelty satelliittipaikannustietoon perustuvaa järjestelmää, joskin sellainen on todennäköisesti tulossa kehityksen edetessä. Solupaikannuksen tarkkuus on kuitenkin tällaisenaan riittämätön ensisijaisena paikannustapana hätäilmoitusten käsittelyssä, mutta se toimii lisätyökaluna silloin, kun paikkatietoa ei saada muutoin tarkasti määriteltyä. (Hätäkeskuslaitos, s.a.) UCL-projektin paikannustarpeisiin solupaikannuksen sen sijaan ajateltiin olevan tarpeeksi tarkka. Toteutusvaihtoehdoista on kerrottu tarkemmin luvussa 5.

4.4 Lähipaikannus

Verkko- ja satelliittipaikannuksessa ei päästä sisätiloissa riittävän tarkkoihin tuloksiin. Niinpä langattomien verkkojen yleistyttyä kodeissa, toimistoissa, hotelleissa, kaupunkien keskustoissa ja muissa julkisissa tiloissa on lähiverkkopaikannuksesta muodostunut potentiaalinen vaihtoehto lähiverkon päätelaitteille. (Rainio, 2003, s. 8)

Lähiverkkopaikannuksen lisäksi muita lähipaikannuksen menetelmiä ovat Rainion (2003, s. 8) mukaan

- Bluetooth-paikannus
- etätunnistinpaikannus
- infrapunapaikannus
- ultraäänipaikannus
- GPS-pseudoliittipaikannus

Näistä tässä työssä käsitellään tarkemmin vain lähiverkkopaikannus, sillä UCL-hankkeen toteutusta varten otettiin selvää, mitä mahdollisuuksia on paikantaa käyttäjä pelkän lähiverkon avulla. Lähiverkkopaikannuksella tarkoitetaan langattoman lähiverkkoyhteyden, WLAN:n, avulla tapahtuvaa paikannusta. Tässä otetaan tämän lisäksi tarkasteluun myös IP-osoitteeseen perustuva paikannus, sillä sitä käytetään UCL-verkkoympäristössä karkean tason sijainnin arvaukseen.

WLAN-paikannuksen periaatteet ovat lähellä solupaikannuksen menetelmiä, sillä kumpikin voi perustua kolmiomittaukseen, tukiaseman tunnistukseen tai signaalin voimakkuuksien mittauksiin. (Paikannus.com, s.a.-a) Tukiasematunnistusta voidaan myös kutsua *sormenjälkitekniikaksi* (engl. fingerprinting), joka perustuu etukäteen mitattuun signaalivoimakkuuskarttaan. Suomalaisen Ekahaun⁵ järjestelmä perustuu tähän. Mobiililaitteen alueella olevat tukiasemat ja niille mitatut signaalivoimakkuudet välitetään palvelimelle, joka puolestaan hakee parhaiten tulosta vastaavan sijainnin kartalta ja palauttaa kyseiset koordinaatit mobiililaitteeseen. Tämän avulla tukiasemien sijaintia ei tarvitse tuntea, mutta jos niiden paikkaa vaihdetaan, täytyy signaalin voimakkuudet mitata uudestaan. (Virtanen & Koskinen, 2005, s. 50-51) Mittaamiseen ja WLAN-verkkojen vianselvitykseen kehitellään uusia, entistä monipuolisempia laitteita (Peltoniemi, 2010), joten tulevaisuudessa lähiverkkopaikannukselle on varmasti entistä enemmän kysyntää. WLAN-paikannus toimii parhaiten sisätiloissa, ja jos mittaustuloksia on useilta eri tukiasemilta ja ympäristön vaikutukset laskettu mukaan, voidaan WLAN-paikannuksella päästä jopa allen metrin tarkkuuteen. WLAN-yhteydet toimivat parhaiten vain pienellä alueella, joten niistä ei ole suureksi uhkaajaksi satelliittipaikannukselle. Voidaankin sanoa, että WLAN-paikannus on paikannusta täydentävä tekniikka siellä missä GPS ei toimi kunnolla eli syvällä sisätiloissa. (Paikannus.com, s.a.-a)

Lähiverkoissa paikannus voidaan tehdä myös IP-osoitteen avulla. Monet sivustot käyttävät sitä tarjoamaan käyttäjilleen muun muassa paikallissään, -uutiset ja -tapahtumia. (Ronda & Bila, s.a., s. 1) Sivustoilla näkee usein myös enemmän ja vähemmän tarpeellisia mainoksia käyttäjän paikkakuntaan tai kotimaahan liittyen. IP-osoite on kuin puhelinnumero, sillä jokaisella on omansa (Viestintävirasto, s.a.). IP-osoitteeseen perustuvaa paikannusta voidaankin verrata luvussa 4.3 esiteltyyn solupaikannukseen. Pystyäkseen kommunikoimaan Internetissä laitteella täytyy olla yksilöllinen IP-osoite, jonka perusteella Internetin paketit löytyvät perille ja laite saa vastauksia takaisin. Samaan tapaan matkapuhelinverkossa puhelimen täytyy olla yhteydessä vähintään yhteen tukiasemaan ja vähintään yhteen sen soluista, jotta matkapuhelinliikenne kulkee puhelimen ja verkon välillä. Näissä kummassakin paikannustarkkuus on huono, mutta menetelmä on yksinkertainen. IP-osoite tai solun numero ei sisällä itsessään paikkatietoa, vaan paikannus tehdään kerättyjen tieto-

⁵<http://www.ekahau.com>

kantojen avulla. Solupaikannuskirjastoja esitellään tarkemmin luvussa 4.6.

Myös IP-paikannukselle löytyy lukuisia sivustoja, kuten IP2Location⁶, IPcatcher.net⁷ ja IP Location Finder⁸. Ongelmana on, että mahdollisia IP-osoitteita on yli 4 miljardia, joten kaikkien ylläpito on mahdotonta. Lisäksi laitteiden IP-osoitteet voivat vaihdella hyvinkin lyhyellä aikavälillä. Apuna on kuitenkin IP-osoitteen jakaminen verkon osoitteeksi (alkuosa) ja verkon sisäiseksi osoitteeksi (loppuosa). Osien pituudet voivat vaihdella, mutta yleisesti niistä voidaan päätellä esimerkiksi, minkä maan, operaattorin tai yrityksen osoitteisiin se kuuluu. IP-paikannuksen tulos voidaan saada myös laitteen *isäntänimestä* (engl. hostname), joka voi esimerkiksi olla 87-94-xyz-xyz.tampere.customers.dnainternet.fi (jossa xyz on luku väliltä 0-255). Paikannustarkkuus voi olla myös lähimmän operaattorin reitittimen mukainen. Näiden tietojen perusteella IP-paikannuksella saadaan yleensä kaupunkikohtainen paikannustarkkuus. Laitekohtainen paikannus IP-osoitteen avulla on kuitenkin käytännössä mahdotonta, sillä esimerkiksi yritysverkon sisällä osoitteet voivat jakautua edelleen yrityksen tietoturvapoliittikan ja laitteiden ominaisuuksien vuoksi miten tahansa. (Ronda & Bila, s.a., s. 1)

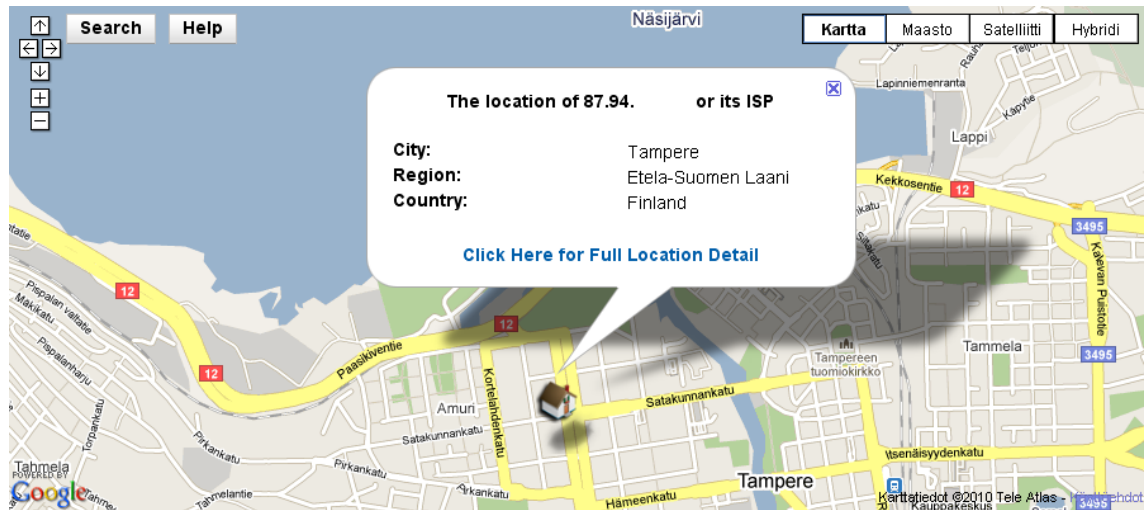
Kuvassa 4.2 on kaupunkikohtainen paikannustulos IP Location Finderin avulla saatuna. Paikannettavana laitteena on pöytäkone, joka on kytketty langallisesti lähiverkkoon. Vastaavanlainen tarkkuus saadaan myös kahdella muulla edellä mainitulla IP-paikannussivustolla. Ainut ero on, että tulos vaihtelee sivustosta riippuen siten, että kohde asetetaan Hämeenkadulle tai sen lähetyville. Tämä johtuu saatujen koordinaattien pyöristyskäytännöstä eri sivustojen välillä. Tällä menetelmällä saadaan Tampereen alueella paikannustulos samaan pisteeseen riippumatta siitä, missä tietokone todellisuudessa sijaitsee. Paikannustarkkuus saattaa siis heitellä kymmenillä kilometreillä. UCL:n tarpeisiin IP-paikannus on kuitenkin hyvä minimivaatimus; kaupunkikohtaisen tuloksen ansiosta kohteen liikuttaminen oikeaan paikkaan on näin helpompaa kuin pelkän maapallonäkymän tapauksessa. Lisäksi IP-osoite saadaan helposti selvitettyä selaimen kautta luvussa 3.2.1. esitetyin menetelmin ilman, että käyttäjän tarvitsee selvittää IP-osoitetta ja syöttää se itse.

On kuitenkin syytä huomata, ettei menetelmä sovi matkapuhelinverkkoon, jossa IP-osoite voi vaihdella mobiililaitteessa jopa muutamien minuuttien välein. Satojen kilometrien päässä toisistaan olevat puhelimet voivat jakaa saman IP-osoiteavaruuden. Tämä tarkoittaa, että osoitteen alkiosa, kuten kaksi ensimmäistä numerosarjaa (16 bittiä), voi kahdella toisistaan täysin erillään olevalla mobiililaitteella olla sama. Tällöin IP-paikannus voi sijoittaa puhelimet samalle alueelle, vaikka todellisuudessa ne olisivat aivan toisella puolella maapalloa tai päinvastoin. Niin-

⁶<http://www.ip2location.com>

⁷<http://www.ipcatcher.net>

⁸<http://www.iplocationfinder.com>



Kuva 4.2: IP Location Finderin paikannustulos Google Mapsin näkymässä

pä IP-paikannuksesta ei ole hyötyä mobiililaitteiden tapauksessa. Lisäksi jatkuvasti vaihtuva IP-osoite on haitaksi esimerkiksi pääsynvalvonnalle, joka voi perustua IP-osoitteen mukaisesti sisällön ja pääsyn suodatukseen. Balakrishnan et al. (2009) testasivat maksullista IP-analysointisivustoa, jonka demoa voi kokeilla maksutta yhdellä IP-osoitteella rajoitetun määrän kertoja päivässä. Matkapuhelimella demon käyttö onnistui helposti lukuisia kertoja peräkkäin, sillä laitteen IP-osoite vaihtui useaan kertaan, esimerkiksi sammutettaessa ja käynnistettäessä Internet-yhteys uudelleen. Samalla testattiin IP-paikannusta yhdelle matkapuhelimelle noin 5-10 minuutin yhteyssykleillä. Paikannus sijoitti matkapuhelimen joka kerta eri puolelle Yhdysvaltoja. (Balakrishnan, Mohamed, & Ramasubramanian, 2009) UCL:n työryhmä kokeili niin ikään IP-paikannusta puhelimelle, jolloin esimerkiksi Tampereen alueella paikannus johtikin Helsinkiin ja pariin kertaan päädyttiin kartalla myös ulkomaille asti.

4.5 Satelliittipaikannus

Rainio (2003, s. 9) erottelee satelliittipaikannuksessa seuraavat menetelmät:

- GPS-paikannus
- differentiaalinen GPS (DGPS, Differential GPS)
- reaaliaikainen kinemaattinen GPS (RTK GPS, Real Time Kinematic GPS)
- GPS-pseudoliittipaikannus
- avustettu GPS-paikannus (A-GPS, Assisted Global Positioning System)

Satelliittipaikannuksessa on kysymys paikannussatelliittien lähettämästä ratatieto- ja aikamerkkisignaalin vastaanotosta ja vastaanottimen sijainnin laskemisesta satelliittien etäisyyksien perusteella. Yli 30-vuotiaan amerikkalaisen GPS-järjestelmän

rinnalle on tulossa viivästyksistä kärsinyt eurooppalainen Galileo-järjestelmä, josta on lisää luvussa 4.5.3. Venäläisillä on oma satelliittipaikannusjärjestelmänsä nimeltään GLONASS (Globalnaya Navigatsionnaya Sputnikova Sistema). (Rainio, 2003, s. 9; Miettinen, 2006, s. 23) Myös kiinalaisilla on oma järjestelmänsä, joka tunnetaan nimeltään Beidou tai COMPASS (GPS Daily, 2010). Tässä työssä keskitytään perinteisen GPS-järjestelmän lisäksi avustettuun GPS-paikannukseen ja Galileo-paikannusjärjestelmään.

4.5.1 Global Positioning System

GPS:n eli Global Positioning Systemin historia sai alkunsa 1973, jolloin Yhdysvaltain puolustusministeriö teki päätöksen uuden järjestelmän tilaamisesta. Järjestelmästä käytetään nimeä Navigation System using Timing and Ranging (Navstar) Global Positioning System (GPS). Sen tavoitteena oli kehittää aseiden suuntaamista ja joukkojen ohjaamista ympäri maailman. Kokonaisuuden hahmottamiseksi GPS-järjestelmän osat voidaan jakaa kolmeen osaan (Miettinen, 2006, s. 32-33):

- Avaruusosa, satelliitit - Aktiivisessa toiminnassa on 24 satelliittia sekä varalla neljä satelliittia.
- Valvontaosa, maa-asema - Keskusasema valvoo ja ohjaa järjestelmän toimintaa sekä tekee tarvittavia korjauksia ja päivityksiä. - Viisi maa-asemaa seuraa tapahtumia taivaalla ja kerää tietoja keskusasemille - Kolme maa-antennia välittää tiedot satelliitteihin.
- GPS-paikantimet - Kymmeniä sovelluksia kaikenlaisten käyttäjien tarpeisiin.

Järjestelmään kuuluu kaikkiaan 28 satelliittia, joista 24 on operatiivisessa käytössä ja loput 4 varmistavat sen, että avaruuteen muodostettu verkko on jatkuvasti aukoton. Satelliittien tarkalla sijoittelulla eli satelliittikonstellaatiolla saavutetaan useita etuja. Tarkkojen ratatasojen avulla taataan, että joka puolella maapalloa olisi aina riittävän monta satelliittia käytettävissä, ja samalla tarkoin suunnitellulla sijoittelulla on säästetty rahaa. Satelliitit kiertävät maata tarpeeksi kaukana toisistaan ja riittävän korkealla horisontin yläpuolella. Napa-alueilla ei saavuteta yhtä tarkkaa geometriaa kuin mantereilla ja valtamerillä, sillä niillä ei ollut Yhdysvalloille samanlaista talous- ja sotilaspolitiikan painoarvoa kuin Neuvostoliitolla. (Miettinen, 2006, s. 33-34)

Jokaisessa satelliitissa on neljä atomikelloa, joita tarvitaan mittaamaan aikaa äärimmäisellä tarkkuudella. Kaksi kelloista laskee aikaa cesium-atomin ja toiset kaksi rubidium-atomin värähtelytaajuuden eli aallonpituuden avulla. Energiatasoissa tapahtuvat muutokset kertovat ajanlaskun, sillä tiettyä energiamuutosta eli hyppäystä vastaa aina saman taajuinen säteily. Cesium-kellossa atomit värähtelevät noin 9

miljardia kertaa sekunnissa ja virhettä ajanlaskuun kertyy korkeintaan yksi sekunti kolmessa miljoonassa vuodessa. Satelliiteissa on lisäksi luotettava radiolähetin ja vastaanotin, joiden avulla GPS-paikantimen tiedonsiirto toimii. Radiosignaali on jaettu kahdelle taajuusalueelle; sotilaiden käyttöön on varattu vähän yli 1200MHz:n taajuus ja siviileiden GPS-laitteet käyttävät noin 1600MHz:n taajuutta. Satelliitit lähettävät maapalloa kohti radiosignaaleja sekunnin tuhannesosan välein. Nämä signaalit sisältävät kantoaaltoon liitetyn eli moduloidun tietosisällön ja koodin. C/A-koodin (Coarse Acquisition Code) avulla järjestelmän ylläpitäjä eli Yhdysvaltojen ilmavoimat voi kontrolloida GPS:n tarkkuutta. Lisäksi koodin ansiosta GPS-paikannin kykenee erottamaan kaikki satelliitit toisistaan, vaikka ne käyttävätkin jatkuvasti samaa taajuutta. (Miettinen, 2006, s. 36-38)

Signaalien tieto-osa on 1500 bittiä pitkä ja se lähetetään joka 30. sekunti varsin hitaalla, 50 bitin sekuntinopeudella (Miettinen, 2006, s. 32-33; Rainio, 2003, s. 9). Doppler-ilmiö aiheuttaa heikolle signaalille poikkeaman taajuuteen, mikä tarkoittaa, että oikea taajuus on "löydettävä" ennen kuin vastaanotto voi alkaa (Rainio, 2003, s. 9). Viestit sisältävät tiedon ilmakehän ylimmän kerroksen muutoksista ja maasemien päivittämät tiedot satelliittien sijainneista avaruudessa sekä tiedon siitä hetkestä, jolloin signaali lähti avaruuteen kohti GPS-paikantimen antennia. Tieto-osasta GPS-paikannin saa myös satelliiteilta kiertoratojen matemaattiset mallit ja almanakat eli ennustetiedot, jotka tunnetaan myös nimeltä efemeridit. Näitä tietoja laite tarvitsee, jotta se osaa pitkien käyttövälien jälkeen hakea horisontin yläpuolelta oikeiden satelliittien tunnistetiedot ja kyetä lukkiutumaan niiden radiosignaaleihin. Tämän ansiosta varmistutaan siitä, että horisontin yläpuolella on aina vähintään kolme satelliittia, jotka GPS vaatii toimiakseen. Tyypillisesti Suomen taivaalla on vähintään 7-8 satelliittia ja parhaimmillaan niitä voi olla 12. GPS-paikantimet ovat nykyisin 12-kanavaisia, joten ne kykenevät ottamaan vastaan ja käsittelemään samaan aikaan kaikista käytettävissä olevista satelliiteista tulevaa tietoa. Tämä on nopeuttanut huomattavasti laitteiden käynnistymistä ja parantanut paikannustarkkuutta sekä lisännyt toimintavarmuutta katveissa. (Miettinen, 2006, s. 34-35, 38)

Vaikka GPS on jo lähes 30 vuotta vanha järjestelmä, niin vasta toukokuusta 2000 lähtien GPS:n paikannustarkkuus on ollut siviilikäytössä suunnilleen nykyisellä tasolla. Vappuna 2000 Yhdysvaltain puolustusministeriön tekemä päätös vaikutti merkittävästi GPS-laitteiden käyttöön ja käyttäjien turvallisuuteen. Tuolloin presidentti Bill Clintonin määräyksestä suljettiin siviilipaikantimien tahallinen häirintä eli SA (Selective Availability). Syynä olivat lähinnä hätäpuheluiden paikannusvaatimukset. SA on atomikellojen manipulointimenetelmä, jossa luodaan merkittävä kellovirhe ja huononnetaan satelliittien ratatietoja. Näitä Coloradon keskusasemalla luotuja kellovirheitä ja ratatietojen virheellisyyttä voidaan säätää huonompaan suuntaan tarpeen mukaan. SA:n poistuttua käytöstä GPS:n käyttäjät havaitsivat vä-

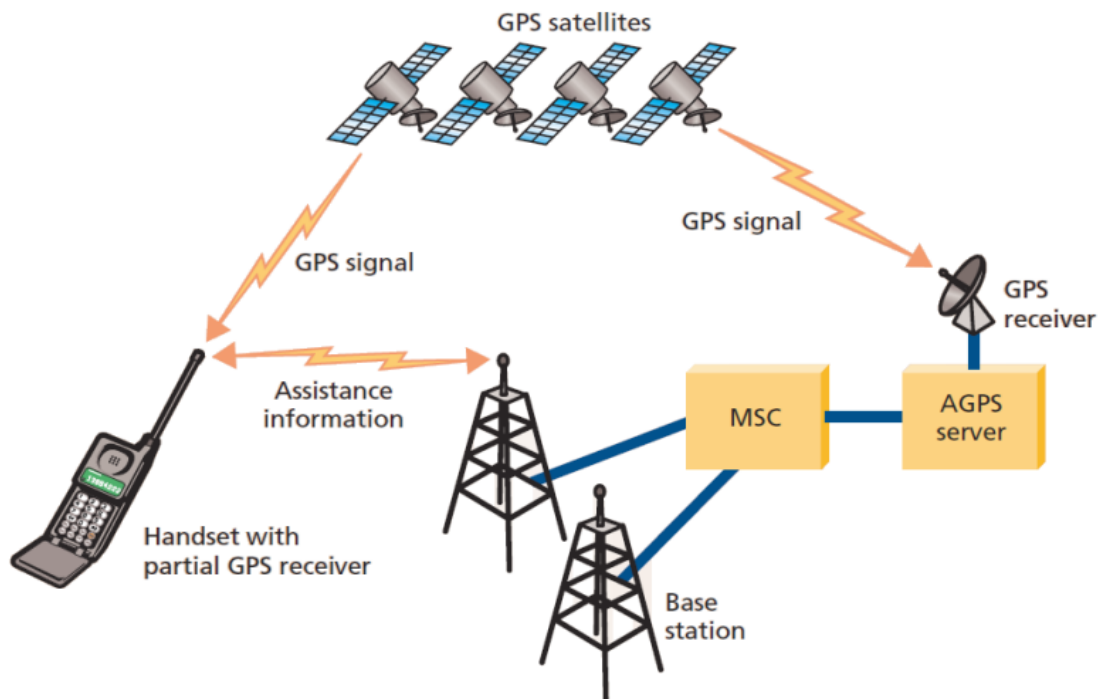
littömästi huomattavia parannuksia paikannustarkkuudessa. Miettinen (2006) vertaa SA-häirintää television viritysnappulaan, jota Yhdysvaltojen ilmavoimat ruuvailee mielensä mukaan. Kun television ruudulla on niin sanottua lumisadetta ja väriseviä juovia, virhettä on hyvin paljon, jolloin GPS:n tarkkuus on huono. Vastaavasti kuvan ollessa terävä ja kirkas myös paikannustarkkuus paranee selvästi. Paikannustarkkuudesta ennen SA:n sulkemista on saatavilla ristiriitaista tietoa, sillä tahallisen häirinnän muodostama virhe vaihteli paljon. Miettinen (2006, s. 48) kirjoittaa, että tarkkuutta saattoi kuvata epämääräisesti toteamalla GPS:n pystyvän parhaimmillaan 30-50 metrin tarkkuuteen. Virhettä saattoi olla hetkellisesti olla 100, 200 tai pahimmillaan 500 metriä. Penttinen (2006, s. 133) puolestaan toteaa, että ennen vuotta 2000 tavallisilla kansalaisilla oli mahdollisuus ainoastaan noin 100 metrin tarkkuuteen. Muutoksen jälkeen tarkkuus parani radikaalisti, sillä tavalliset paikantimet pystyivät yllättäen 5-10 metrin tarkkuuteen. Virheitä voi edelleenkin ilmetä, mutta jo 30-40 metrin tarkkuus luokitellaan huonoksi. (Miettinen, 2006, s. 48-49) Merkittävimmät virhelähteet ovat ilmakehän hidastava vaikutus signaaliin, kello- ja ratavirheet, paikantimien omat virheet, satelliittien keskinäisien sijaintien vaihtelut sekä monitieheijastukset. (Miettinen, 2006, s. 52-59)

Miksi sitten GPS ei toimi sisätiloissa? Joskus GPS-signaalin löytäminen on hankalaa myös autoissa tai veneissä. Tämä johtuu siitä, että GPS-yhteys vaatii esteettömän yhteyden taivaalle. Satelliiteista tulevat radiosignaalit eivät nimittäin pääse etenemään kiinteiden rakenteiden, auton peltien tai veneen katteiden läpi. Radioiden alhainen lähetysteho ja käytetty taajuusalue ovat syinä tähän. Hetkellisiä katveita, joissa GPS-paikannus ei toimi, voi syntyä peitteisessä maastossa, kaupunkialueilla tai pysäköintihalleissa ja tunneleissa. GPS jatkaa kuitenkin toimintaansa normaalisti, kun siirrytään avoimemmalle paikalle. Hetkellisiä katveita on pyritty poistamaan liittämällä GPS-paikantimeen kiihtyvyyteen reagoivia antureita, mikä tarkoittaa, että satelliittipaikannukseen yhdistetään niin kutsuttu inertia-paikannus. Katvealueet ovat tosin pienentyneet pääosin radiovastaanottimien herkistämisen seurauksena. Lisäksi myös edellä mainittu 12-kanavaisuus on vaikuttanut tähän. Tämän ansioista vastaanottimet kykenevät toimimaan paremmin rakennusten sisällä, kunhan lähettyvillä on ikkuna, sillä GPS-signaalit läpäisevät hieman vaimentuneina ikkunan tai tuulilasin. (Miettinen, 2006, s. 37)

4.5.2 Assisted GPS

GPS:n suurin ongelma on siis paikantaminen erityisesti sisätiloissa, jonne heikon signaalin on vaikea edetä. Matkapuhelimeissa GPS ei ole yleistynyt kovinkaan nopeasti, mikä todettiin luvussa 4.1. Lisäksi GPS-puhelimeissa on ongelmana laskentatehon vaatima suuri virrankulutus. Lisäksi GPS:n hidas käynnistyminen (tästä käytetään myös termiä kylmäkäynnistys) on ongelma nopeaa paikkatietoa tarvitsevalle. Ongel-

mien seurauksena on kehitetty matkapuhelinverkon avustama satelliittipaikannus eli A-GPS (Assisted GPS), jonka toiminto näkyy kuvassa 4.3. Matkapuhelin saa ensin verkosta satelliittien rata- ja mahdolliset korjaustiedot, joiden perusteella se valitsee satelliitit, joita kannattaa käyttää. Käytännössä valinta kohdistuu niihin satelliitteihin, jotka sijaitsevat samalla puolella maapalloa kuin puhelin. Tämän jälkeen puhelin tekee laskentatyön itse, kuten tavanomaisessa GPS-paikannuksessa, mutta paikkatieto saadaan huomattavasti nopeammin. (Paikannus.com, s.a.-c) Tämän puhelimesta tapahtuvan paikannuksen (handset-based) lisäksi on myös mahdollista, että laskenta tapahtuu verkon palvelimessa (network-based). (Rainio, 2003, s. 11) Tällöin paikkatiedon laskentatyö lähetetään erillisen paikannuspalvelimen tehtäväksi, mikä säästää puhelimen virtaa, mahdollistaa niin ikään nopean paikantamisen ja paremmasta laskentatehosta johtuen mahdollisesti tarkemman tuloksen. Normaalin GPS-paikannukseen voi kulua kymmeniä sekunteja, kun taas A-GPS paikantaa jopa muutamassa sekunnissa. (Rainio, 2003, s. 11; Nokia, s.a.-a) Etuna on myös se, että A-GPS toimii kaikkialla maailmassa operaattoreiden palveluista riippumatta, kunhan puhelimella on yhteys matkapuhelinverkkoon. Tiedonsiirtoon käytetään 2G- ja 3G-yhteyttä, eikä A-GPS ainakaan vielä tue WLAN-yhteyttä. Yhteyden siirtämä datamäärä ei ole kovinkaan iso, tyypillisesti alle 10 kilobittiä. (Nokia, s.a.-a)



Kuva 4.3: A-GPS:n toimintaperiaate (Djuknic & Richton, 2001)

Nopea paikantaminen huomattiin myös UCL-työryhmän omissa testeissä, joissa A-GPS antoi puhelimeen paikkatiedon käytännössä välittömästi sitä pyydettyä.

Samalla havaittiin vaihteluita paikannustarkkuudessa eri tiloissa. Tampereen teknillisen yliopiston sisätiloissa A-GPS paikansi puhelimen Hervannan valtavyylälle, jonne matkaa kertyy todellisesta sijainnista noin 100 metriä riippuen, missä kohdin rakennusta paikannusta käytetään. Sisätiloissa tarkkuuden luvataan olevan noin 20-30 metriä heikon signaalin vuoksi. Ulkotiloissa paikannustarkkuus ei parantunut merkittävästi, joten A-GPS:lle luvattu noin 10 metrin tarkkuus ulkotiloissa jäi melko kauaksi. Jopa 2-3 metrin tarkkuus pitäisi olla mahdollinen. (Paikannus.com, s.a.-c) Niinpä projektin testeissä ei päästy luvattuihin tuloksiin, mutta A-GPS:n näkyvin hyöty eli erittäin nopea paikantaminen lisäsi mielenkiintoa kehittää erillinen, koordinaatit hakeva sovellus puhelimiin.

4.5.3 Galileo-paikannusjärjestelmä

Keväällä 1999 Euroopan Avaruusjärjestö ESA:aan kuuluvien 14 maan liikenneministerit päättivät eurooppalaisen satelliittipaikannusjärjestelmän rakentamisesta. Myöhemmin nimetyn Galileon tekniikan tarkoitus oli irtaantua Yhdysvalloista poliittisesti, sotilaallisesti ja taloudellisesti. Hankkeen kehitystyötä varten varattiin 235 miljoonaa euroa, ja myös Suomi oli mukana päätöksenteossa. Galileo on pääasiassa tarkoitettu siviilien käyttötarpeisiin. Sen tavoitteena on luoda kansallisista ja sotilaallisesta tarpeista irrotettu järjestelmä, jonka avulla voidaan tuottaa palveluita muun muassa erilaisten kuljetusjärjestelmien tarpeisiin. Täysin eurooppalainen hanke ei kuitenkaan tarkoita, että Galileo kilpailisi amerikkalaisen GPS:n kanssa. Hankkeen edetessä EU:n ja Yhdysvaltojen välit ovat asian suhteen parantuneet niin, että nämä ovat voineet sopia yhteistyöstä. Yhdysvallat on suostunut nykyaikaistamaan GPS:ää yhteensopivaksi Galileon kanssa niin, että se ottaa käyttöön uudempaa, mutta jossain määrin myös riskialttiimpaa tekniikkaa. Vaikka nykyiset GPS-laitteet eivät voi ottaa vastaan Galileon signaalia, on tarkoitus, että markkinoille tulee paikantimia, jotka voivat hyödyntää kumpaakin järjestelmää. Lisäksi nykykäyttäjien helpotukseksi on sovittu, että GPS tulee kaikella varmuudella toimimaan myös tulevaisuudessa. (Miettinen, 2006, s. 182,184)

Galileo on kärsinyt viime vuosien aikana selviä viivästyksiä. Vuosituhannen alussa suunnitelmana oli, että sen täysi operatiivinen toiminta olisi tarkoitus saavuttaa vuoteen 2008 mennessä. (Rainio, 2003, s. 12) Myöhemmin arvioksi esitettiin vuotta 2010. (Miettinen, 2006, s. 183) Uusimpien tietojen mukaan EU-komissio uskoo nyt, että järjestelmän toiminta alkaa vuonna 2014. Samalla myös hankkeen kustannukset ovat nousseet. Alun perin hankkeen piti maksaa EU:lle noin 1,8 miljardia euroa, mutta yksityisen ja kaupallisen omistuksen malli epäonnistui. EU rahoittaa koko hankkeensa itse ja tämä tulee arviolta maksamaan 4-5 miljardia euroa. Vuoden 2010 aikana on suunnitelmissa tehdä vielä uusia sopimuksia, sillä järjestelmään tarvitaan vielä 4-12 satelliittia sekä maanpäällisiä ohjausjärjestelmiä. Satelliitteja

tulee yhteensä olemaan vähintään 22 ja enimmillään alkuperäisten suunnitelmien mukaisesti 30. (Kotilainen, 2010)

Järjestelmän valmistuessa suunnitellusti se tulee koostumaan kaikkiaan 30 satelliitista, joista 27 on jatkuvassa käytössä ja 3 on varasatelliitteina. Järjestelmällä on kolme lentorataa, joilla kullakin on kymmenen satelliittia. Tavoitteena on tuottaa hieman parempi peitto maapallon pohjoisilla alueilla kuin GPS:llä. Niinpä erityisesti Suomi hyötyisi Galileon peittoalueesta. Huolellisesti suunniteltujen kiertoratojen ansiosta paikannussignaali on otettavissa paremmin vastaan kuin nyt ja myös korkeiden rakennusten keskellä. Myöskään maan vaihtelevat muodot eivät Galileo-järjestelmässä aiheuta katveita, kuten nyt tapahtuu. Lisäksi järjestelmän tueksi rakennetaan laaja maa-asemien verkko, joka välittää järjestelmälle viestejä muun muassa tarkkuuden arvioinnista ja laitteiston viosta. Euroopan komissio on ehdottanut Galileolle eri palvelumuotoja, joista *Open Service* (OS) on erityisesti sovelluskäyttäjille mielenkiintoinen maksuttomuutensa ja kilpailukykynsä takia. Sitä ei tosin suojata häiriöiltä tai toimintakatkoksilta. Tarkkuudeksi on kaavailtu 5-20 metriä. *Commercial Service* (CS) on puolestaan selvästi tarkempi, jopa 0,1-10 metriä, ja siihen liittyy suojaus ja maksullinen laatutakuu. *Safety of Life* (SoL) on myös suojattu ja sopii esimerkiksi ilmailuun ja merenkulkuun, joissa ihmishenkien suojaaminen on tärkeitä. Palvelu varottaisi käyttäjiä, jos satelliittien ilmoittaman paikannustarkkuuden ollessa syystä tai toisesta epäluotettava. Neljäs palvelumuoto olisi edellisiä paremmin suojattu ja tarkoitettu vain turvallisuusviranomaisien käyttöön. Tämän *PRS-osion* (Public Regulated Signal) käyttöoikeutta ei myönnettäisi kenelle tahansa viranomaiselle, vaan käyttöä valvottaisiin ja se olisi luvanvaraista. PRS:n laadukas suojaus takaisi, että sitä voitaisiin käyttää esimerkiksi kriisialueilla. Tarkkuudeksi sille suunnitellaan 1-6 metriä. Viides ja viimeinen luokka on *Search and Rescue Service* (SAR), jolla on mahdollista välittää hälytysviestejä, jotka vastaanotetaan hälytysjärjestelmistä. Tämä on samalla laajennus kansainväliseen COSPAS-SARSAT-etsintä- ja pelastuspalvelujärjestelmään⁹. (Miettinen, 2006, s. 183-184; Rainio, 2003, s. 12; Penttinen, 2006, s. 136)

Satelliittipaikannuksen tulevaisuuden näkymät ovat siis erittäin hyvät, kunhan Galileo saadaan toimimaan suunnitellusti. Miettinen (2006, s. 185) toteaa yleisesti satelliittipaikannuksen olevan erinomainen apuväline ja sen tuottaman tiedon olevan tarkkaa myös vaativissa olosuhteissa. Tästä huolimatta se ei saisi koskaan syrjäyttää perinteisten paikantamisen ja suunnistamisen menetelmiä ja välineitä, maastotai merikartan lukutaitoa, paperille painettua karttaa ja magneettikompassia. Huolena on muun muassa sähköjärjestelmän sortuminen ja se, että maailmaa ollaan rakentamassa yhä riippuvaisemmaksi avaruudessa kiertävistä satelliiteista ja niistä saatavasta tiedosta.

⁹<http://en.wikipedia.org/wiki/COSPAS-SARSAT> [viitattu: 8.3.2010]

4.6 Solupaikannuskirjastot

Solupaikannus vaikutti sopivalta vaihtoehdolta UCL-projektin tarpeisiin, sillä se on yksinkertainen paikannusmenetelmä, johon löytyy useampi eri toteutusvaihtoehto. Internetissä on lukuisia websivustoja, joihin on kerätty tukiasemasolujen sijainteja ja näiden avulla muodostettu solutietokantoja, joita tässä kutsutaan solupaikannuskirjastoiksi. Suurin haaste on saada solutieto puhelimesta näkyviin, sillä tiedon arkaluonteisuuden vuoksi sitä ei löydy niin vain esimerkiksi puhelimen valikoista.

Valmiita ohjelmia haettaessa löytyi erittäin lupaavalta vaikuttava CellTrack¹⁰, joka on tarkoitettu S60-puhelimille. Kun ohjelman käynnistetään, se hakee matkapuhelinverkon signaaleihin liittyviä tietoja, kuten solun numeron, signaalin voimakkuuden, maakoodin, operaattoritunnuksen ja operaattorin verkon aluekoodin sekä tiedon, onko käytettävä verkko 2G vai 3G. Lisäksi soluja on mahdollista hakea ja tallentaa valmiiseen tietokantaan, jonne saa myös kuvia. Ainakaan Tampereen alueelta ei oletetusti löytynyt tietokannasta mitään, mutta ohjelma antoi joka kerta hyvin nopeasti yhteyteen liittyvät tiedot, joista tärkein on siis solun numero. Niinpä solutieto on helposti saatavilla ohjelman avulla, mutta ongelmana on löytää kyseisen solun sijainnin koordinaatit. UCL:ssä tätä voitaisiin hyödyntää niin, että kun käyttäjä saa koordinaatit, hän syöttäisi ne sellaisenaan UCL:n verkkoympäristöön. Sijainti voitaisiin näin esittää kartalla Google Mapsin¹¹ avulla.

4.6.1 OpenCellID

OpenCellID¹² on avoimeen lähdekoodiin perustuva projekti, jonka tavoitteena on luoda maailmanlaajuinen solutunnistetietokanta paikkatietoineen. Paikkatietoja voi tietokannasta hakea sivuston rajapinnan¹³ avulla tai ladata koko solutietokanta¹⁴ itselleen. Lisäksi solujen sijainteja voi tutkia sivustolla yksitellen Google Mapsin karttanäkymän avulla tai ladata kaikki solut maakohtaisesti .kml-tiedostona, jota tukee Google Earth¹⁵. Tämän tiedoston voi linkittää myös Google Mapsiin sijoittamalla karttihakukenttään tiedoston sijainti. Kuvassa 4.4 näkyy palvelun tarjoamat Tampereen keskustan solut Google Mapsin karttanäkymässä. Kaikkiaan soluja tietokannassa on yli 600000 ja pelkästään Suomessa soluja on 8212 (tiedot haettu 1.3.2010). Alkuun sivusto vaikutti erittäin aktiiviselta, mutta Suomen osalta soluja ei ole tullut kuin noin 100 lisää sitten heinäkuun 2009.

CellTrackin kanssa palvelu toimii hyvin, kunhan puhelimen käyttämä solu löytyy

¹⁰<http://www.afischer-online.de/sos/celltrack>

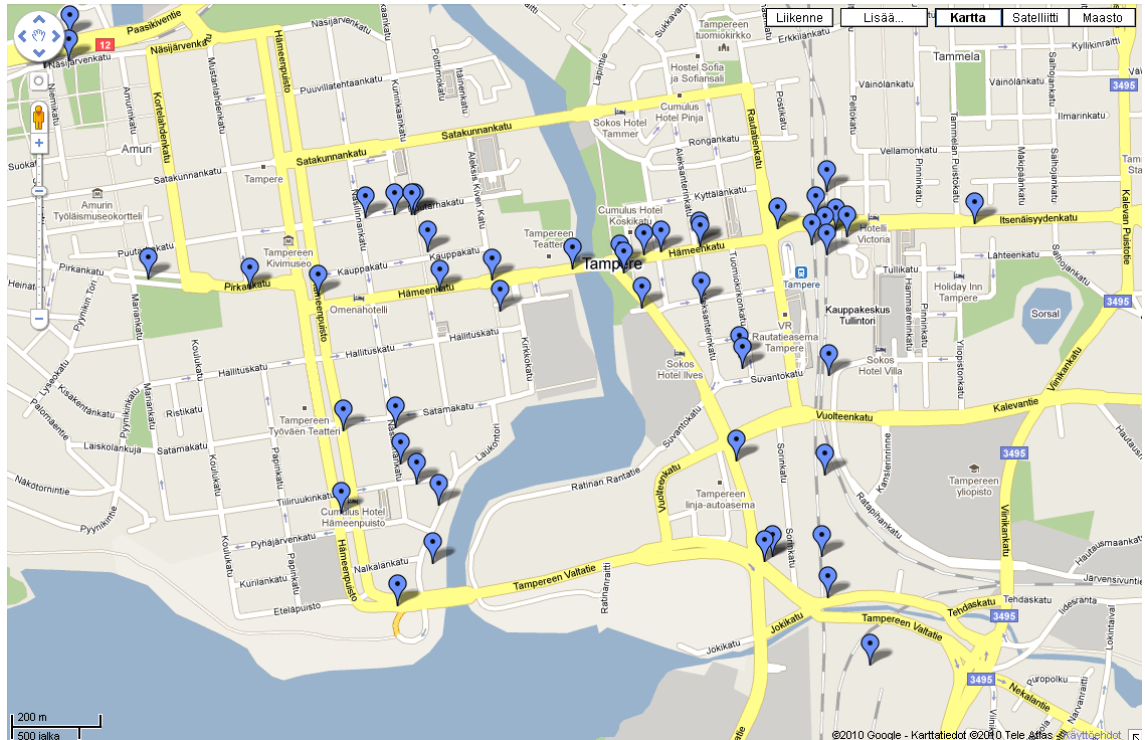
¹¹<http://maps.google.fi>

¹²<http://www.opencellid.org>

¹³<http://www.opencellid.org/api>

¹⁴<http://myapp.fr/cellsIdData>

¹⁵<http://earth.google.com>



Kuva 4.4: OpenCellID:n tukiasemasolut Tampereen keskustassa

tietokannasta. Esimerkiksi testissä käytetty Nokian etäpuhelin palauttaa solunumeron 22440 maatunnukseksi 244 ja operaattoritunnukseksi 05. Nämä syöttämällä OpenCellID API:n ohjeiden mukaisesti URL-tietoihin¹⁶ sivusto palauttaa listauksen 4.1 mukaiset tiedot.

Listaus 4.1: OpenCellID:n koordinaatit XML-dokumentissa solunumerolle 22440

```

1 <cell range="50000" lac="29121" lat="60.2208320808249"
2 nbSamples="76" lon="24.7204207253429" cellId=""
3 mcc="244" mnc="5"/>

```

Tästä saadaan koordinaatit poimittua esimerkiksi XML-jäsentimellä ja ne voidaan siirtää Google Mapsiin¹⁷. Samaan tapaan voi lisätä tukiasemasoluja tietokantaan. Sivustolle rekisteröitymisen jälkeen jokainen käyttäjä saa niin kutsutun API-avaimen (apikey), joka sijoitetaan osoitetietoihin solutietojen lisäksi. Soluja ei ole pakko syöttää yksitellen, vaan useamman solun saa ladattua sivustolle CSV-muodossa¹⁸.

Palvelun ongelma on se, että jos matkapuhelimen käyttämää solua ei löydy, sivusto palauttaa lähimmän ja joissain tapauksissa satunnaisen solun koordinaatit.

¹⁶<http://www.opencellid.org/cell/get?cellid=22440&mcc=244&mnc=5&lac=29121>

¹⁷http://maps.google.com/maps?f=q&source=s_q&hl=en&geocode=&q=60.220783876012,+24.720556031203&sll=60.852227,24.704819&sspn=0.028344,0.077162&ie=UTF8&ll=60.220868,24.722006&spn=0.007876,0.024333&z=16

¹⁸<http://www.opencellid.org/measure/upload>

Tulos saattaa siis epäonnistua useilla kymmenillä kilometreillä. Suomen mittakaavassa noin 8000 solua ei riitä kattamaan kuin osittain suurimmat kaupungit. Niinpä palvelun monipuolista käyttöä varten täytyisi hakea ja syöttää itse esimerkiksi Hervannan solutiedot, jotta OpenCellID:n käyttö onnistuisi paremmin esimerkiksi TTY:n ympäristössä.

4.6.2 CellID Novay

CellID Novayn¹⁹ tarjoamassa tietokannassa on puolestaan yli 132000 solua ja suomalaisia soluja on 14262 (tieto poimittu 1.3.2010). Palvelusta saa ladattua OpenCellID:n tapaan solujen sijainnit KML-tiedostona. Tampereen alueella soluja on kymmeniä enemmän kuin OpenCellID:n tietokannassa. Kartalla näkyy myös arvio solujen peittoalueista, mikä jättää allensa muun muassa kadunnimet ja häittää näin luettavuutta. Sivustolla voi lisäksi hakea tietokannasta solujen sijainteja²⁰. Pyydetty tiedot eli maakoodi, operaattorikoodi, aluekoodi ja solun numero saadaan CellTrac-kista. Samoilla tiedoilla kuin OpenCellID:n tapauksessa tietokanta palauttaa koordinaatit ja tarjoaa suoran linkin kyseisillä koordinaateilla Google Mapsiin²¹. Listauksia 4.1 ja 4.2 vertailtaessa huomataan, että koordinaatit eivät ole täysin samat. Kartalla ero on pari kilometriä, mikä selittyy sillä, että CellID Novayn tulos on saatu vain aluekoodin tasolla, kuten listauksessa 4.2 ilmoitetaan. Tietokannassa on tuloksia myös solutarkkuudella, jolloin virhearvioksi ilmoitetaan noin 340 metriä. Uusia soluja voi tietokantaan lisätä IYOUIT-ohjelman²² avulla. Vaatimuksena on S60-puhelin joko sisäisellä tai ulkoisella GPS-vastaanottimella.

Listaus 4.2: CellID Novayn koordinaatit haetuilla tiedoilla (maakoodi eli MCC 224, operaattorikoodi eli MNC 5, aluekoodi eli LAC 29121, solun numero eli CID 22440)

-
- ₁ This cell 22440 is located at (60.2449865,24.7324445) +/- 9 km
₂ Location could be determined on area code level.
-

Hakuja tietokantaa voi tehdä vain kymmenen päivässä, mutta rekisteröitymällä on mahdollista suorittaa hakuja rajattomasti. Listauksen 4.2 mukaisesti sijaintitietoja ei palvelusta kannata hakea, sillä tällä tavoin hakuja ei voi automatisoida. Rajapintoina tietokantaan on tarjolla Java ja .NET, joiden käyttöönotto tosin nähtiin liian työläänä. Lisäksi palvelu ei tuo tarpeeksi lisäarvoa verrattaessa OpenCellID:een, vaikka soluja Suomessa onkin selvästi enemmän.

¹⁹<http://cellid.novay.nl>

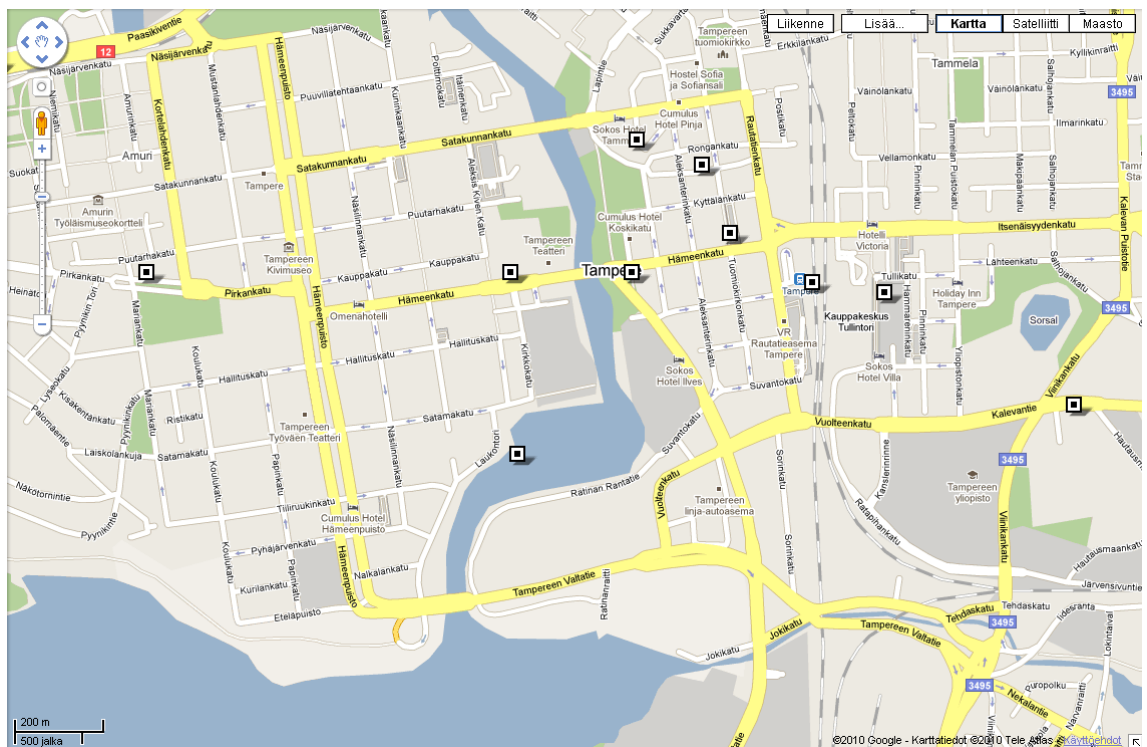
²⁰<http://cellid.novay.nl:8002/cellid/Request.aspx>

²¹<http://maps.google.fi>

²²<http://www.iyouit.eu/portal>

4.6.3 CellSpotting

Myös CellSpotting²³ on kahden edellisen tapaan sivusto, jonne on kerätty tukiasemasolujen sijainteja, kaikkiaan soluja on yli 111000. Maakohtaisia lukuja sivustolla ei ole saatavilla, mutta tietokantaa²⁴ selatessa huomaa, että suomalaisia soluja on useita satoja, joskin kahden edellisen solupaikannuskirjaston lukuihin ei päästä. Tämä nähdään myös kuvasta 4.5, jossa on solut Tampereen keskustassa. Tämän lisäksi suurin osa soluista on lisätty 2004 tai 2005, joten tiedot eivät välttämättä pidä enää paikkaansa. Erona edellisiin solupaikannuskirjastoihin on se, että CellSpotting tarjoaa oman ohjelmansa matkapuhelimille. Näin ollen CellTrackia ei välttämättä tarvitse käyttää, sillä ohjelmalla näkee sen hetkisen solun tiedot. Jos solua ei tietokannasta löydy, on se mahdollista tallentaa suoraan ohjelman kautta. S60-puhelinten lisäksi sivustolla on oma versionsa Sony Ericssonin JP-7 ja JP-8 puhelimille, joten puhelintuki on melko kattava.²⁵ Sen sijaan käytettävyyttä heikentää vähäisten solujen määrän lisäksi se, että solujen numerot ovat ASCII-muodossa. Myöskään mitään kirjastoa ei ole tarjolla, joten paikkatietojen automatisoitu kerääminen ei onnistu.



Kuva 4.5: CellSpottingin tukiasemasolut Tampereen keskustassa

²³<http://www.cellspotting.com>

²⁴<http://www.cellspotting.com/thick/browse.php>

²⁵<http://www.cellspotting.com/software>

4.7 Geolocation API

Mozillan Firefoxin Geolocation-toiminto²⁶ kertoo verkkosivulle selaimen sijainnin. Toiminnosta käytetään nimitystä *sijaintitietoinen selaus* (engl. Location-Aware Browsing), ja esimerkkinä sen hyödyntämisestä mainitaan lähimmän pizzerian selvittäminen. Toinen esimerkki on karttapalvelu, jolle riittää kertoa minne käyttäjä on menossa, sillä sijainti eli lähtöpaikka on toiminnon avulla tiedossa. (Mozilla, s.a.) Toiminto vaatii joko Firefox-selaimen (versio 3.5 tai uudempi), Seamonkey-selaimen (versio 2 tai uudempi), Thunderbird-sähköpostilukuohjelman (versio 3 tai uudempi) tai Google Chrome -selaimen (5.0 beta tai uudempi) (Mozilla Developer Center, 2010; Lardinois, 2010). Myös Applen iPhoneen Safari-selain tukee tätä toimintoa (Etherington, 2009), joten samalla on mahdollista sijaintitieto ainakin yhdestä matkapuhelinmallista. Lisäksi voidaan olettaa, että tuki Geolocation API:lle tulee ajan myötä myös muihin selaimiin, sillä se on W3C:n tukema. W3C:n työryhmä on määritellyt turvallisen ja yksityisyyden huomioivan rajapinnan palvelun käytölle. (W3C, s.a.)

Käyttäjälle toiminto näkyy siten, että sijaintitietoisella verkkosivulla Firefox-selain kysyy lupaa käyttäjän sijaintitiedon jakamiseen. Jos lupa annetaan, toiminto kerää seuraavia tietoja (Mozilla, 2010):

- päätelaitteen sijainti sisäisen tai ulkoisen GPS-laitteen kautta
- Internet-yhteyden palveluntarjoajan toimittama sijainti
- lähimpien WiFi-reitittimien sijainnit
- lähimpien matkapuhelinverkon tukiasemien solunumerot
- lähimpien langattomien lähiverkkojen päätelaitteiden liityntäpisteiden (Wireless Access Point) ja/tai matkapuhelinverkon tukiasemien signaalin voimakkuudet
- päätelaitteen IP-osoitteen

Kerätyt tiedot välitetään paikannuspalvelimen oletustarjoajalle, Google Location Servicesille, joka tekee ja palauttaa arvioinnin käyttäjän sijainnista. Tämän jälkeen sijaintitieto, käytännössä leveys- ja pituusaste, välitetään kyselyn tehneelle verkkosivulle, jolla lopputulos voi näkyä esimerkiksi käyttäjän sijoittamisena kartalle. Google, Mozilla tai Mozillan palvelimet eivät kerää sijaintitietoja, eikä myöskään Firefox-selain muista käyttäjän sijaintia. Selain tosin muistaa satunnaisen Googlen tarjoaman käyttäjätunnisteen, joka uusiutuu kahden viikon välein. Jos puolestaan

²⁶<http://www.mozilla.com/fi/firefox/geolocation>

toiminnoille ei anneta lupaa, niin selain ei tee mitään. Toiminnon käyttäminen on siis täysin vapaaehtoista, sillä Firefox ei välitä sijaintitietoja ilman käyttäjän lupaa. Myös toiminnon käyttöönoton luvataan olevan helppoa, sillä paikannustoiminto noudattaa avoimia standardeja. (Mozilla, 2010; Google Code, s.a.)

Sijaintitietojen tarkkuus vaihtelee merkittävästi riippuen siitä, käytetäänkö päätelaitteessa langallista vai langatonta lähiverkkoyhteyttä. Palveluntarjoajien toimitamat sijainnit ovat lisäksi vain arvioita, joten selvitettyjen sijaintien tarkkuutta ei voida taata. Kun päätelaite on kytketty WLAN-yhteydellä, Google Location Services osaa kertoa päätelaitteen käyttämän tukiaseman sijaintitietojen avulla käyttäjän sijainnin. Tällöin paikannustarkkuus voi olla jopa muutama metri eli se on verrattavissa satelliittipaikannuksen tarkkuuteen. Päätelaitteen ollessa kytkettynä LAN-yhteydellä paikannus perustuu tavalliseen IP-paikannukseen, jolloin käyttäjän sijainnin mukaan merkataan lähimmän kaupungin tai keskustan keskipiste. Tarkkuus vaihtelee myös sijainnista riippuen, sillä pienillä paikkakunnilla IP-osoitteen perusteella tehty paikannus saattaa tarjota sijainniksi kymmenien kilometrien päässä olevan kaupungin. Samoin, jos päätelaitteen käyttämän langattoman tukiaseman sijaintitietoja ei ole saatavilla, tehdään paikannus epätarkemman eli IP-paikannuksen mukaisesti. (Mozilla, s.a.; Mozilla, 2010)

4.8 Paikannusta hyödyntäviä palveluita

Lukuisia paikkatietoisia palveluita on ilmestynyt viime vuosina markkinoille. Syitä tähän ovat älypuhelimien yleistymisen (etenkin GPS-vastaanotin ja WLAN), mobiiliyhteyksien yleistymisen (mobiililiittymä eli moka kiinteällä kuukausimaksulla) sekä GPS-vastaanottimien määrän kasvu ja sen vuoksi hintojen aleneminen ja laitteiden yleistymisen. Erityisesti ruokapaikkoihin ja liikuntaan liittyvien paikannuspalveluiden määrä on kasvanut selvästi, mutta myös henkilöiden paikkatietoja sisältäviä palveluita on ilmestynyt useampi. Tässä luvussa esitellään muutamia paikkatietoa hyödyntäviä palveluita, jotka tulivat vastaan UCL-hankkeen paikannusvaihtoehtoja ja -esimerkkejä haettaessa. Palvelut ovat myös innostaneet etsimään ratkaisuja käyttäjien paikannukseen, sillä osa toiminnoista on ratkaisuiltaan mielenkiintoisia ja niillä on selvästi motivoiva vaikutus käyttäjiin.

Prinssi Edwardin saaren yliopiston²⁷ uudistettu mobiiliverkkosivusto sisältää mielenkiintoisen ominaisuuden liittyen paikkatiedon hyödyntämiseen puhelimissa. Yliopisto on merkannut kampuksensa kartalle²⁸ kaikki alueen rakennukset. Jokainen rakennus näkyy osoittimena Google Maps²⁹ -näkyvässä. Tämän ansiosta GPS-vastaanottimella varusteltu puhelin, kuten Apple iPhone, näyttää kampusalueelle,

²⁷<http://www.upei.ca>

²⁸<http://www.upei.ca/facilities/interactive-campus-map>

²⁹<http://maps.google.fi>

kuinka paljon on metreissä matkaa eri rakennuksiin. (Drupal, 2009) Tästä lienee eniten hyötyä etenkin uusille opiskelijoille ja vierailijoille, joille yliopiston rakennusten sijainnit eivät ole tiedossa. Yliopiston mukaan tavoitteena on mahdollistaa jopa henkilöiden paikannus niin, että voidaan rakennuksen perusteella nähdä ihmiset, jotka ovat rakennuksessa ja tiettyä henkilöä etsiessä nähdä, missä rakennuksessa hän on.

Yksi esimerkki käyttäjien paikkatietoa hyödyntävästä palvelusta on Baeo³⁰. Sivuston FAQ:n mukaan palvelun avulla voi nähdä kartalla kavereiden, perheenjäsenten tai työkavereiden nykyiset tai viimeisimmät sijainnit. Palveluun voi myös merkata eri sijainteja ja sen avulla voi ottaa yhteyttä vanhoihin kavereihin. Palvelun käyttö ei vaadi GPS-vastaanotinta, vaan matkapuhelin paikannetaan pelkästään tukiasemien solunumeroja ja signaalien voimakkuuksia käyttäen. Tarkkuuden luvataan olevan kaupunkialueiden noin 300 metrin ja maaseutujen neljän kilometrin välillä. Baeoon tukiasemasolujen tietokantaa voivat mobiilikehittäjät hyödyntää palvelun määrittämien rajapinnan avulla. (Apostolakis, 2008) Baeoon keräämät paikkatiedot (kaupunki, osavaltio ja kunta) saa myös automaattisesti vietyä Twitter-verkkopalveluun³¹. Palveluun ja sen rajapintaan ei päästy UCL-projektin puitteissa tarkemmin tutustumaan, sillä vaikka palvelun käyttö onkin ilmaista, rekisteröinti ja palvelun käyttö edellyttävät Windows Mobile -matkapuhelinta.

Yksi motivoivimmista paikannuspalveluista projektin aikana oli Nokian Sports Tracker Beta³², joka nimensä mukaisesti kerää tietoja käyttäjien urheilu suorituksista. Palvelu on ilmainen ainakin vielä Beta-vaiheessa ja sen avulla voi monitoroida, analysoida ja jakaa muille omia suorituksiaan. Puhelimeen asennettava ohjelma tallentaa automaattisesti kuntopäiväkirjaan tietoja, kuten nopeuden, matkan pituuden ja ajan. Kerättyä dataa voi visualisoida ja palvelun avulla voi käydä suoritusta uudestaan läpi. Dataa voi myös jakaa muille 2G- tai 3G-verkon kautta ja on myös mahdollista kilpailla itseään tai kaveriaan vastaan. Palvelu toimii Nokian S60-puhelimissa ja se vaatii joko sisäisen tai ulkoisen GPS-vastaanottimen. (Nokia, s.a.-b) Sports Tracker oli siinä mielessä motivoiva UCL:ään liittyvä palvelu, sillä siitä huomasi, miten paikkatietoa voidaan parhaimmillaan hyödyntää. Palvelu näytti konkreettisesti, että puhelimesta on todellakin mahdollista saada paikkatieto talteen suoraan verkkosivustolle reaaliaikaisesti. Myös Sports Tracker -palvelun motivoiva vaikutus sen käyttäjiin näkyy selvästi. Palkittu ja suosittu palvelu toimii eräänlaisena henkilökohtaisena valmentajana esimerkiksi juoksuharrastukselle. Joitakin käyttäjiä voi motivoida erilaiset tilastotiedot, toisia puolestaan oman kehityksen seuraaminen yleisemmällä tasolla. Myös sillä, että voi nähdä urheilu suorituksensa konkreettisesti kartalla, on motivoiva vaikutus palvelun käyttöön.

³⁰<http://www.baeeo.com>

³¹<http://twitter.com/baeeo>

³²<http://sportstracker.nokia.com>

Suomalaista osaamista edustaa Nokian lisäksi Gemilo Oy:n toteuttama ilmainen Mapsi³³, joka näyttää kartalla käyttäjän lähellä olevat kohteet, kuten nähtävyydet, ravintolat, terassit, kahvilat ja tapahtumat. Yleiskarttana toimii OpenStreetMap³⁴, ja kaupunkikarttoina on saatavana Turun ja Tampereen keskusta-alueet. (Mapsi, s.a.) Mapsin toteutukseen harkittiin Google Mapsin³⁵ käyttöä, mutta sen käytön arveltiin tulevan liian kalliiksi, sillä Mapsiin on tarkoitus toteuttaa myös maksullisia osioita ja ominaisuuksia. (Mapsi, 2009) Mapsi-palvelussa kohteet näkyvät kartalla eri tunnuksina, ja jos kohde ei ole avoinna, se on merkattu harmaalla. Kohteiden valikoista voi soittaa suoraan esimerkiksi ravintolaan ilman, että tarvitsisi näppäillä numeroita puhelimeen. Kohteita on myös mahdollista kommentoida, mikä helpottaa käyttäjiä sopivien kohteiden valitsemisessa. Mapsin paikannustekniikkaa tai -tarkkuutta ei tarkkaan kerrota, mutta kuvauksen perusteella voi olettaa, että paikannus perustuu GPS-paikannukseen. Käyttäjän tulee nimittäin antaa lupa käyttää paikannusta ja puhelimen on tuettava sijaintitietoja. (Mapsi, s.a.) Turun ja Tampereen keskustojen kohteiden lisäksi Mapsi on toteuttanut mobiilin karttapalvelun myös Tampereen yliopistolle. Maaliskuun 2010 alussa avattu palvelu toimii yliopiston keskuskampuksella ja sen tavoitteena on helpottaa erityisesti pääsykokeisiin tulevien, uusien opiskelijoiden sekä liikuntaesteisten henkilöiden liikkumista alueella. Karttaan on merkattu muun muassa yhteiset opetustilat, ravintolat, infopisteet ja kirjasto. Liikuntaesteiset on huomioitu merkkamalla myös hissit, automaattiovet ja inva-wc -tilat. (Tampereen yliopisto, 2010)

Tässä mainittiin siis vain merkittävimmät ja kiinnostavimmat paikannuspalvelut, jotka tulivat vastaan esimerkkejä ja motivointia haettaessa. Lisää esimerkkejä paikannusta hyödyntävistä palveluista olisi tarjolla muutamia tuhansia. Paikannusohjelmien räjähdysmäistä kasvua on luvattu jo tämän vuosituhannen alusta alkaen, mutta vasta parina viime vuonna niiden määrä on kasvanut huomattavasti. Vuoden 2010 on arveltu olevan viimein paikannuksen vuosi, jolloin kyseiset palvelut yleistyvät räjähdysmäisesti. (Malik, 2010; Nelson, 2010) Kuten aikaisemmin jo todettiin, suurin syy tähän on GPS:n yleistyminen matkapuhelimissa. Paikannuspalveluiden kasvu ei kuitenkaan rajoitu vain matkapuhelimiin, sillä esimerkiksi kuljetus- ja työmaakoneyritykset ovat hyödyntäneet ja tulevat jatkossakin hyödyntämään yhä enemmän paikannusta. Hyöty näkyy niiden toiminnassa, sillä lähetysten ja koneiden paikantamisella voidaan optimoida toimintaa ja parantaa logistiikkaa. Tästä hyötyvät myös viranomaiset ja asiakkaat, sillä tiedetään reaaliajassa, missä kohteet ovat. (Sipilä, 2010; Nelson, 2010) Matkapuhelinsovelluksia ajatellen sosiaalisen median palvelut ovat osoittaneet, että ihmiset haluavat jakaa tietojansa niille,

³³<http://www.mapsi.fi>

³⁴<http://www.openstreetmap.org>

³⁵<http://maps.google.fi>

joihin he luottavat. Tämän myötä paikkatiedon jakamisen kynnys on madaltunut. (Nelson, 2010) Niinpä myös UCL:n tapauksessa paikkatieto koettiin erittäin mielenkiintoiseksi ominaisuudeksi osana käyttökokemuksia. Arveltiin, että ominaisuuden motivoivuus saattaisi vaikuttaa merkittävästi palvelun käyttöön. Samalla haluttiin yksinkertaisesti nähdä, miten valmiita ihmiset ovat kertomaan sijaintinsa tietyllä ajanhetkellä.

5. TAPAUSKUVAUS: UCL

UCL:n toteutusta varten karotettiin siis, miten matkapuhelimesta ja selaimesta saataisiin UCL-verkkoympäristöön laitetiedot sekä miten ja kuinka tarkasti on mahdollista saada paikkatieto talteen. Tässä luvussa esitellään UCL-verkkoympäristö yleisesti ja miten päätelaitetiedot ja paikkatieto kerätään UCL:ssä.

5.1 UCL-verkkoympäristön esittely

Urban Computing Labin toteutuslueksi valittiin Drupal¹, joka on PHP-pohjainen avoimen lähdekoodin *sisällönhallintajärjestelmä* ja *sovelluskehitysalusta*. Sen ovat valinneet käyttöönsä useat suomalaiset sivustot, kuten Nokia Siemens Networks², Sanoma Entertainment³, Suomi24⁴ ja YLE⁵ (Palmu, 2008). Drupalin käyttöönottoa UCL:n alustaksi puolsivat useat sen ominaisuudet, kuten tuki TTY:n käyttämälle Shibboleth-kirjautumiselle⁶. Sen avulla TTY:n käyttäjätunnuksilla voi kirjautua ja luoda käyttäjätilin suoraan Drupalilla toteutettuun verkkosivustoon. Tämä nopeuttaa kirjautumista UCL:ään ja vähentää tarvetta luoda kokonaan uudet, muista palveluista täysin erilliset tunnukset sivustolle. Samalla tiedostettiin, ettei Shibboleth-kirjautuminen toimi matkapuhelimella, joten myös tavanomainen kirjautuminen täytyi toteuttaa. Drupalin valintaa tuki myös luvussa 3.6 esitellyt moduulit, jotka tuottivat toivottuja tuloksia erillisessä testiympäristössä, joten nämä valmiit työkalut haluttiin osaksi laitetunnistusta. Drupaliin kehitellään jatkuvasti uusia moduuleja ja entisiä päivitetään kehittäjien aktiivisuuden mukaan, joten mahdollisten uusien ominaisuuksien lisääminen verkkoympäristöön onnistuu helposti. Myös erilaiset teemat ja visualisointityökalut vaikuttivat valintaan. Osittain samaan aikaan Hypermedialaboratoriossa toteutettiin TTY-Piiri⁷, joka pohjautuu niin ikään Drupaliin.

UCL toteutettiin TTY-Piirin tapaan Hlab-palvelimelle ja sen verkko-osoitteesta (<http://hlab.ee.tut.fi/ucl>) haluttiin mahdollisimman yksinkertainen, jotta se on helppo kirjoittaa myös mobiililaitteilla. Samasta osoitteesta on siis mobiililaitteille tarkoitettu versio. Sivustoa avatessa päätelaitteen tiedot luetaan Mobile Plugin

¹<http://drupal.org>

²<http://www.nokiasiemensnetworks.com>

³<http://www.sanomaentertainment.com>

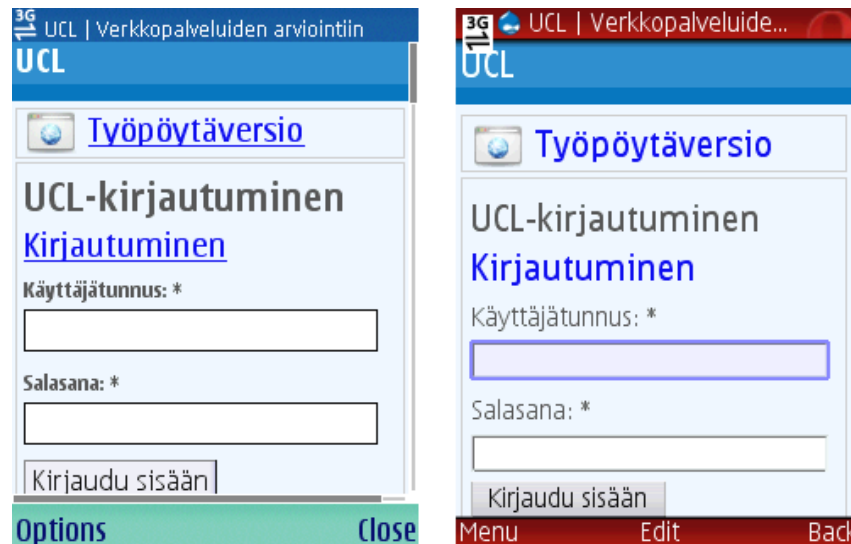
⁴<http://suomi24.fi>

⁵<http://www.yle.fi>

⁶<http://shibboleth.internet2.edu>

⁷<http://hlab.ee.tut.fi/piiri>

-moduulilla, jonka testituloksia on esitelty liitteen 1 esimerkeissä 1-3. Mikäli pääte-laite tunnistetaan matkapuhelimeksi (esim. Nokia), vaihdetaan samalla moduulilla sivuston käyttämää tyyli-tiedostoa, jolloin etusivun näkymä on paremmin optimoitu matkapuhelimille. Tyyli-tiedoston valinta perustuu puhelimen merkin ja mallin sijasta puhelimen käyttämään selaimen. Nokian oman selaimen ja Opera Minin tiedettiin olevan käytetyimmät mobiiliselaimet, joten näitä varten luotiin omat tyyli-tiedostot. Kummankin selaimen etusivu näkyy kuvassa 5.1.



Kuva 5.1: UCL-verkkoympäristön etusivu N73:ssa Nokian ja Opera Minin selaimessa

Jos mobiiliselainta ei puolestaan tunnisteta, käytetään moduulin kehittäjän tekemää Mobile Garland -teemaa⁸. Jokainen matkapuhelimelle tarkoitettu ulkoasu ottaa myös huomioon kosketusnäyttölliset puhelimet muun muassa suurentamalla linkkejä ja niiden kuvakkeita, jotta niihin olisi helpompi osua. Lisäksi muita kuin matkapuhelimia varten tehtiin oletustyyli-tiedosto, joka vastaa sivuston tavanomaisesta työpöytänäköymästä kuvan 5.2 mukaisesti.

Tyylien lisäksi myös tiettyjä ominaisuuksia on karsittu mobiiliversiosta, jotta se toimisi nopeammin. Esimerkiksi matkapuhelimille hyödytön Shibboleth-kirjautuminen on poistettu etusivulta. Samoin työpöytänäköymän vasemman laidan linkit on piilotettu, jotta ne eivät veisi tilaa pienillä näytöillä. Koska sivuston navigoinnista haluttiin tehdä mahdollisimman yksinkertainen ja selkeä, avuksi on otettu kuvakkeita linkkien yhteyteen. Erityisesti mobiilinäköymässä on painotettu selkeä kirjautumista ja käyttökokemuksien tallentamista. Ajatuksena on, että mobiililaitteilla vain tallennetaan käyttökokemuksia UCL:ään. Tavanomaisilla selaimilla voidaan puolestaan kätevämmän muokata omia käyttökokemuksia, selata palveluita, joista käyttökokemuksia on tehty ja nähdä erilaisia visualisointeja käyttökokemuksista. Samoin ohjeen selkeyteen ja käytettävyyteen matkapuhelimilla on kiinnitetty

⁸http://drupal.org/project/mobile_garland



Kuva 5.2: UCL-verkkoympäristön etusivu Firefox-selaimessa

huomiota, jotta käyttökokemuksen kirjaaminen olisi mahdollisimman selkeätä.

5.2 Käyttökokemuksen tallentaminen

UCL:n käyttöönoton helpottamiseksi on tehty tarkat ja mahdollisimman selkeät ohjeet⁹, jotta verkkoympäristön käyttö olisi mahdollisimman helppoa, eikä palvelun opettelukynnys olisi korkea. Käyttökokemuksen kirjauksessa täytetään lomake, joka on pituutensa vuoksi jaettu neljään eri kokonaisuuteen. Ideana on ollut luvussa 2 esitellyt viisi eri tasoa (yleinen, fyysinen, sosiaalinen, toiminnallinen ja affektiivinen), joita on sovellettu UCL:n tarpeiden mukaisesti. Käyttökokemuksen nimeämisen jälkeen lomakkeella kysytään käytetyn verkkopalvelun taustatietoja, joita ovat palvelun nimi ja osoite sekä käyttötarkoitus, käyttötiheys ja hyödyllisyys. Seuraavaksi täytetään käyttötilanteeseen liittyvät tiedot, joita ovat verkkopalvelun käytön päivämäärä, kesto, kyseisen päivän mieliala ja sosiaalinen käyttötilanne eli käytettiinkö palvelua yksin, jonkun kanssa vai jouduttiinko kysymään apua. Lisäksi käyttäjä voi syöttää avainsanoja eli tageja käyttöön vaikuttavista tekijöistä, käyttöpaikasta ja sijainnista. Näistä avainsanoista muodostetaan käyttökokemusten visualisointien yhteydessä *tagipilviä* (engl. tag cloud¹⁰), joista voi tarkastella avainsanojen mukaisesti esimerkiksi, millaisia palveluita käyttäjät ovat arvioineet. Näiden lisäksi käyttäjältä kysytään sijaintia, jossa arvioitavan verkkopalvelun käyttö tapahtui. Sijainnin voi syöttää joko katuosoitteena, paikkakuntana ja maana tai koordinaatteina. Koordinaatit voi kirjoittaa suoraan niille tarkoitettuihin kenttiin tai siirtää Google Map-

⁹<http://hlab.ee.tut.fi/ucl/miten-kaytan-uclaa>

¹⁰http://en.wikipedia.org/wiki/Tag_cloud [viitattu: 24.4.2010]

sin¹¹ karttanäkymässä osoitinta sijaintinsa mukaisesti. Nopein ja kätevin vaihtoehto on käyttää Geolocation API:a, joka esitellään luvussa 4.7. Lisää paikannuksen toteutuksesta on luvussa 5.3.

Sijainnin jälkeen kysytään käyttäjältä palvelun käytön onnistumiseen liittyvät tiedot eli tavoitteen saavuttaminen, käyttökokemuksen tunnetila, arvosana ja kommentteja. Lopussa täytetään vielä tekniset tiedot päätelaitteesta, jolla arvioitavaa verkkopalvelua käytettiin. Kaikkien vaihtoehtojen (pöytäkone, kannettava, miniläppäri, matkapuhelin, muu) kohdalla kysytään selain ja sen versio, yhteystyyppi, IP-osoite sekä näytön resoluutio. Valittaessa pöytäkone, kannettava tai miniläppäri kysytään lisäksi käyttöjärjestelmä ja vastaavasti matkapuhelimen yhteydessä kysytään puhelimen merkkiä ja mallia. Päätelaitetunnistuksesta on kerrottu tarkemmin luvussa 5.4. Vielä ennen käyttökokemuksen tallentamista on mahdollista lisätä tiedosto, kuten kuva tai video, rikastamaan käyttökokemuksen tietoja. Kuva voi esimerkiksi olla käyttötilanteeseen liittyvä tarkennus tai kuvakaappaus arvioitavan verkkopalvelun aiheuttamasta virhetilanteesta päätelaitteessa. Tämän jälkeen käyttökokemus tallennetaan ja siitä lähetetään käyttäjälle muistutuksena sähköposti, jossa on linkki käyttökokemukseen. Lomakkeen tietoja on mahdollista muokata jälkikäteen.

Käyttökokemuslomakkeen nopeampaa täyttämistä varten jokaisen vaihtoehdon kohdalle on sijoitettu ikoni, joka havainnollistaa kysytyjä tietoja ja elävöittää sivua paremmin. Tämä auttaa varsinkin ensimmäisen tallennuskerran jälkeen, sillä jo kuvakkeiden perusteella voi muistella vaihtoehdot. Samoin kosketusnäyttöpuhelimissa voi ikoneita painamalla valita haluansa vaihtoehdon lomakkeelle. Lisäksi kirjausta nopeuttamaan luotiin käyttäjille *sovelluskirjanmerkki* (engl. bookmarklet¹²). Sovelluskirjanmerkki, jota kutsutaan myös nimillä aktiivinen kirjanmerkki ja toimintokirjanmerkki, sisältää ulkoisen URL-osoitteen sijaan kutsun selaimen JavaScript-tulkkiin. Sen avulla voi lisätä selaimen käyttökelpoisempia ja parempia työkaluja, esimerkiksi voi muuttaa sivuston kirjasinkokoa ja taustaväriä tai poimia hiirellä valittu teksti vaikkapa hakukoneeseen. Sovelluskirjanmerkki eroaa muusta JavaScript-koodista siten, että koodi täytyy kirjoittaa yhdelle riville ja peräkkäiset komennot täytyy erotella puolipisteillä. (Negrino & Smith, 2007, s. 419) Työpöytäselaimille, kuten Firefoxille, Operalle ja Safarille tarkoitettun UCL:n sovelluskirjanmerkin sisältö näkyy listauksessa 5.1. Tallentamalla ja avaamalla sovelluskirjanmerkin aukeaa UCL:n käyttökokemuslomake uuteen selainikkunaan. Lomakkeelle poimitaan automaattisesti selaimen nykyisen sivuston nimi ja verkko-osoite sekä mahdollinen valittu teksti kopioidaan kommenttikenttään.

¹¹<http://maps.google.fi>

¹²<http://en.wikipedia.org/wiki/Bookmarklet> [viitattu: 24.4.2010]

Listaus 5.1: Sovelluskirjanmerkin koodi työpöytäselaimille

```

1 javascript :
2 u=document.location.hostname+document.location.pathname ;
3 t=document.title ; s=window.getSelection () ; w=screen.width ;
4 h=screen.height ; void (window.open ("%22http://hlab.ee.tut.fi/ucl/
5 node/add/experience?edit[group_usage][field_service_name][0]
6 [value]=%22+escape(t)+'&edit[group_success][field_comments][0]
7 [value]='+escape(s)+'&address='+escape(u)+'&width='+escape(w)+
8 '&height='+escape(h) , '_blank' , ' width=1024,height=700,
9 status=yes , resizable=yes , scrollbars=yes '))

```

Sovelluskirjanmerkit toimivat rajoitetusti myös matkapuhelinten selaimissa, kuten Nokian omassa selaimessa ja Opera Minissä (Salonen & Huhtamäki, 2010, s. 11). UCL:n tapauksessa sovelluskirjanmerkin koodia jouduttiin muokkaamaan, sillä kyseiset selaimet eivät hyväksy tallennettavaksi kirjanmerkkiä, ellei sen koodi ala URL:n alkuosalla, kuten listauksessa 5.2. Samoin matkapuhelinten selaimet eivät tue muun muassa uuden ikkunan avaamista ja sen koon asettamista, eikä tekstiä voi kaikilla matkapuhelimilla valita verkkosivuilta, joten nämä on karsittu pois. Lisäksi matkapuhelinten selaimet eivät hyväksy heittomerkin käyttöä koodissa, joten ne on korvattava entiteeteillä. Sovelluskirjanmerkin käytöstä on hyötyä erityisesti matkapuhelinta käytettäessä, sillä tällöin arvioitavan verkkosivuston osoitetta ei tarvitse näppäillä käsin lomakkeelle. Samoin useista palveluista saadaan otsikkotiedoista luettua sivuston nimi, joten tämäkin nopeuttaa käyttökokemuksen kirjaamista.

Listaus 5.2: Sovelluskirjanmerkin koodi matkapuhelinten selaimille

```

1 http://hlab.ee.tut.fi/ucl/node?javascript:document.location=
2 %27http://hlab.ee.tut.fi/ucl/node/add/experience?edit
3 [group_usage][field_service_name][0][value]=%27+
4 escape(document.title)+%27&address=%27+escape(
5 document.location.hostname+document.location.pathname)

```

5.3 Paikannuksen toteutus

Luvussa 4.7 esiteltyä Geolocation API:a tukevan selaimen ylälaitaan ilmestyy lomakesivun auetessa pyyntö saada määrittää selaimen sijainti. Pyyntö on muotoiltu niin, että Hypermedialaboratorion palvelin¹³ haluaisi tietää, missä käyttäjä sijaitsee. Todellisuudessa paikannuksen sallimisen jälkeen tiedot lähetetään paikannuspalvelimen oletustarjoajalle, joka on tässä tapauksessa Google Location Service. Niinpä varsinaista paikannusta ei tehdä UCL-verkkoympäristössä. Tämän jälkeen toiminto sijoittaa osoittimen kartalle hakemiensa koordinaattien perusteella. Erityisesti

¹³hlab.ee.tut.fi

WLAN-yhteyttä käytettäessä tulos on tarkka, jopa muutamia metrejä. Langallisen lähiverkon tapauksessa paikannus tehdään IP-osoitteen perusteella, jolla päästään yleensä vain kaupunkikohtaiseen tarkkuuteen, kuten todettiin luvussa 4.4. Jos käyttäjän selain ei tue Geolocation API:a, suoritetaan IP-paikannus [hostip.info](http://www.hostip.info)¹⁴ API:n avulla. Näiden kahden eri IP-paikannustoteutuksen välinen tarkkuus vaihtelee muutamilla sadoilla metreillä kaupungista riippuen. Jos puolestaan automaattinen paikannus sijoittaa käyttäjän väärään paikkaan tai käyttäjä ei halua antaa Geolocation API:n määrittää sijaintia, voi kartalla siirtää osoitinta haluamaansa paikkaan. On myös mahdollista, että käyttäjä tallentaa käyttökokemuksen ajallisesti eri aikaan kuin arvioitavaa verkkopalvelua käytettiin, jolloin sijainti on voinut muuttua. Näitä varten lomakkeelle voi kirjoittaa sijaintinsa katuosoitteen, jolloin Google Maps¹⁵ hakee sijainnin kartalle osoitetietojen avulla.

Geolocation API toimii erittäin hyvin paikantamiseen työpöytäkoneiden selaimissa, varsinkin kun käytössä on WLAN tai matkapuhelinverkko. Puhelimista Iphonella paikannus toimii niin ikään hyvin, sillä Safari-selain tukee ominaisuutta. IP-paikannus puolestaan antaa suuntaa-antavan sijainnin, jota on helppo muokata kartalla. Lomakkeen pienen karttakoon vuoksi kosketusnäyttöpuhelimilla kartan käyttö on vaivalloista. Katuosoitteen mukainen paikannus toimii niin ikään hyvin, mutta ongelmana on, että käyttäjä joutuu aina syöttämään osoitetiedot itse. Suurin osa käyttäjistä kuitenkin tietää, millä kadulla arvioitavaa palvelua käytti, joten siinä mielessä katuosoitteen kysyminen on perusteltua ja tulokset usein riittävän tarkkoja. Matkapuhelinten paikannukselle ei löytynyt ratkaisua yhtä nopeasti ja vaivattomasti, Iphonea lukuun ottamatta, mutta puhelinten paikannusmahdollisuudet haluttiin kuitenkin kartoittaa. Solupaikannuksen tulokset olivat heikkoja ainakin Suomen osalta riittämättömien solupaikannuskirjastojen takia, mutta täysin kyseistä paikannusta ei haluttu sulkea pois. Erillisten ohjelmien käyttömahdollisuus haluttiin selvittää, sillä tiedossa oli, että solun numeron lisäksi esimerkiksi GPS-vastaanottimen kanssa puhelimen koordinaatit on mahdollista saada puhelimesta esille. Tällöin UCL:n käyttäjä voisi syöttää saadut tiedot verkkoympäristössä niille tarkoitettuihin kenttiin tai parhaimmassa tapauksessa puhelin lähettäisi ne suoraan käyttökokemuslomakkeelle.

5.3.1 Java ME

Kehitysalustoista potentiaalisin laitteiden kattavuuden kannalta on Java ME, sillä se tukee sekä Series 40- että S60-puhelimia. Java Platform Micro Edition eli Java ME (aikaisempi nimi on J2ME) on yksi kolmesta suuresta Java-standardin ympäristöstä, johon kuuluvat Enterprise Edition (Java EE) ja Standard Edition (Java

¹⁴<http://www.hostip.info>

¹⁵<http://maps.google.fi>

SE). Java SE on Javan perusympäristö ja se on tarkoitettu ensisijaisesti PC- ja WWW-maailmaan, jossa on käytössä suorituskyvyltään tehokas laitteisto ja paljon muistia. Niinpä standardi tarjoaa laajan kirjastovalikoiman eri tarkoituksiin. Java EE on puolestaan palvelinkäyttöön tarkoitettu Java-ratkaisu, joka on teknisesti kuin Java SE täydennettynä palvelinkäyttöön soveltuvilla kirjastoilla. Java ME on suunniteltu mobiililaitteikäyttöön ja se määrittelee kaksi erilaista ohjelmistokokonpanoa. Monimutkaisempi näistä kahdesta on *Personal Java*, joka on ajoajan, muistin ja suorituskyvyn vaatimuksiltaan hyvin lähellä Java SE:n virtuaalikonetta ja ohjelmistovaatimuksia. Yksinkertaisempi ratkaisu on *Kilo-virtuaalikone* (KVM, Kilo virtual machine), joka ottaa paremmin huomioon heikkotehoiset päätelaitteet. (Mikkonen, 2004, s. 83-84) Java ME on siis luotu mahdollisimman joustavaksi, joten se on kevyt ja se on tarkoitettu rajoitettujen laitteiden, kuten juuri matkapuhelimien, PDA-laitteiden ja hakulaitteiden ohjelmointiin. Java ME:n luokkakirjastoja on karsittu, jotta se soveltuisi paremmin ympäristöön, jossa laitteiden muistia ja muita resursseja on vain rajallinen määrä käytettävissä. Alustalla kirjoitettuja ohjelmia kutsutaan MIDleteiksi. Javaa osaavan on helppo omaksua ME:n ominaisuudet, sillä osa luokkakirjastoista on samoja kuin Java EE:ssä ja Java SE:ssä. Java ME on laajasti levinnyt Java-teknologia, sillä monet suuret valmistajat tarjoavat laitteissaan sille tuen. Niinpä kyseisellä alustalla ohjelmoitu sovellus voi toimia jopa 3 miljardissa laitteessa. (Forum Nokia Library, s.a.; Symbian Foundation, 2010a)

Java ME:n päätelaitepaikannusta varten on olemassa kokonaan oma luokkakirjasto nimeltään Location API for J2ME (JSR 179)¹⁶, joka tukee kymmeniä matkapuhelimia¹⁷. Se sisältää tarvittavat kirjastot paikkatietoisien palvelun toteuttamiseen, joka voi perustua esimerkiksi solupaikannukseen tai GPS-paikannukseen. (Mahmoud, 2004) Nämä samat vaihtoehdot olivat UCL:n paikannusvaihtoehtoina, mutta vaikka toteuttamiselle löytyi runsaasti esimerkkejä¹⁸, MIDletien käytön esti maksulliset sertifikaatit. Java ME:llä kirjoitetut ohjelmat täytyy allekirjoittaa, jotta päätelaitteet hyväksyvät ohjelmien asentamisen. (Forum Nokia Wiki, 2009c) Ohjelmat on mahdollista allekirjoittaa itse ja saada ne siten toimimaan päätelaitteessa, mutta itse allekirjoitetut ohjelmat rajoittavat yhteensopivien puhelinten määrää. Tällöin MIDlet ei toimi Series 40 eikä S60 3rd Edition -puhelimissa. (Forum Nokia Wiki, 2009b) Sama todettiin UCL:n Java ME -kokeiluissa, joissa puhelimesta riippuen oma ohjelma joko asentui tai ei asentunut puhelimeen. Asentuessaan ohjelman paikannusfunktiot palauttivat joko tyhjä arvot tai virheilmoituksen. Täten olisi tarvittu maksullinen sertifikaatti, joka ei tullut hintansa vuoksi kysymykseen. Niitä

¹⁶<http://www.jcp.org/en/jsr/detail?id=179>

¹⁷<http://devices.j2mepolish.org/interactivedb/searchdevices.faces?term=polish.api.locationapi&columns%3Djavaconfiguration%2Cjavaplatform%2Cscreenize&sorting=identifier>

¹⁸http://wiki.forum.nokia.com/index.php/Portal:Java_Code_Examples

olisi täytynyt hankkia ainakin kaksi erilaista, jotta paikannusohjelma asentuisi mahdollisimman moneen matkapuhelimeen (Forum Nokia Wiki, 2009f). Hintaa yhdellä Java-sertifikaatilla¹⁹ on satoja dollareita. Niinpä Java ME:stä ei tullut vaihtoehtoa matkapuhelinten paikantamiselle.

5.3.2 Python for S60

Python for S60 eli PyS60 on nimensä mukaisesti ratkaisu käyttää Python-kieltä S60-puhelimien sovellusten kirjoittamiseen. Lisänä tavanomaiseen Python-kieleen, PyS60 mahdollistaa pääsyn moniin älypuhelimien toiminnallisuuksiin, kuten kameraan, kontakteihin, kalenteriin, nauhuriin, peleihin, puheluihin sekä TCP/IP- ja Bluetooth-yhteyksiin. PyS60 on perustunut avoimeen lähdekoodiin vuodesta 2006 lähtien, jolloin Nokia julkisti sen noin vuoden kehityksen jälkeen avoimeksi. Nokia halusi suositun ja vahvan Python-kielen helpottamaan sovellusten kirjoittamista ja tarjota sen käyttöön vastaavat kirjastot kuin on Symbian C++:ssa. Toimiakseen S60-puhelimeen täytyy asentaa Python Runtime ja Python Script Shell²⁰. (Nokia, 2008, s. 1; Open Source at Nokia, 2010)

PyS60:n etuna on, että se on helppo oppia ja kielen ilmaisuvoima mahdollistaa edellä mainittujen toimintojen lisäksi muun muassa GPS:n käytön. Kielellä kirjoitetut skriptit voi ajaa sellaisenaan emulaattorilla ja puhelimessa Script Shellin avulla ilman, että niitä tarvitsee varsinaisesti asentaa laitteeseen. PyS60:lla on mahdollista käyttää monipuolisempia toimintoja kuin esimerkiksi Javalla tai Flash Litella. Lisäksi avoin lähdekoodi takaa, että kehittäjät voivat muokata sitä vapaasti ja sen työkalut ovat ilmaisia kaikille. PyS60:n negatiivinen puoli on, että se ei tue muita kuin S60-puhelimia ja siinä on vähemmän mahdollisia toiminnallisuuksia kuin natiivissa Symbian C++:ssa. PyS60:n suorituskyky on kohtuullista tasoa, mutta se ei pärjää Symbian C++:lle, joten se ei sovi hyvin esimerkiksi korkealaatuiseen videostreamaukseen. Lisäksi Python Runtime on asennettava käsin puhelimeen, jotta Pythonilla kirjoitetut ohjelmat toimivat. (Symbian Foundation, 2010b) Myös PyS60:n kanssa ilmenee ongelmia, kun sillä kirjoitettua ohjelmaa yritti muuttaa SIS-tiedostoksi (Software Installation Script), joka on asennustiedosto Symbian-käyttöjärjestelmälle, ja allekirjoittaa se. Allekirjoitetun SIS-tiedoston avulla on mahdollista välttää Pythonin Runtimen asennus puhelimeen ja se mahdollistaa ohjelman ajamisen ilman Python Script Shelliä. Puhelinmallista riippuen ohjelma ei joko suostunut asentumaan puhelimeen tai ohjelma palautti paikannuskoordinaateiksi tyhjä arvot. Ohjelmaa keikeltiin myös suoraan Python Script Shellillä, mutta tällöin paikannuskoordinaattien funktio palautti virheilmoituksen allekirjoittamattoman ohjelman käytöstä. PyS60

¹⁹<http://www.thawte.com/code-signing/index.html> ja <http://www.verisign.com/code-signing/content-signing-certificates/index.html>

²⁰<http://sourceforge.net/projects/pys60>

ei siis tuonut ratkaisua oman paikkatiedon palauttavan ohjelman tekemiseen. Vaikka ohjelmien ajo Python Script Shellillä onnistuisi, se vaatii aina käyttäjältä ylimääräistä paneutumista asentamiseen, mikä puolestaan vie aikaa ja vaikuttaa UCL:n käytettävyyteen.

5.3.3 Sovellusesimerkkejä

Java ME:sta ja PyS60:stä ei löytynyt ratkaisua matkapuhelinten paikantamiselle edellä esitettyjen ongelmien takia. Myös muita kehitysalustoja kartoitettiin vaihtoehdoksi paikannusohjelman tekemiseksi. Forum Nokia Wikistä²¹ poimitujen koodiesimerkkien ja oppaiden²² avulla selvitettiin Symbian C++:n, Qt:n, Open C/C++:n, Web Runtimen ja Flash Liten ominaisuuksia. Näiden perusteellisempaa testausta ei toteutettu, sillä Symbian C++:aa lukuun ottamatta kaikkien laitetuki on puuttellista. Suurin osa on tarkoitettu S60-puhelimille ja ainoastaan Flash Lite toimii Java ME:n tapaan sekä Series 40- että S60-puhelimeissa. Lisäksi alustoja oli huomattavasti vaikeampi lähestyä, sillä niiden käyttämistä koodikielistä oli UCL-työryhmällä vähemmän kokemusta kuin Java ME:stä ja PyS60:tä.

Kaikissa kehitysalustoissa kirjastojen käytön hidasteena on niiden arkaluonteisuus. Tämä tarkoittaa ainakin Nokian tapauksessa sitä, että tiettyjä kirjastoja ja funktiota eli ominaisuuksia²³ käytettäessä ohjelmat tulee allekirjoittaa. Allekirjoittamisesta voi tulla käytetyistä ominaisuuksista riippuen monimutkainen prosessi ja kuten aikaisemmin todettiin, paikannuskirjaston käyttöön ei riitä kehittäjän oma allekirjoittaminen. (Forum Nokia, s.a.-a) Niinpä omien ohjelmien lisäksi paikannusratkaisuksi etsittiin samalla valmiita sovelluksia, joiden toimivuus ja monipuolisuus haluttiin nähdä. Lisäksi valmiit sovellukset autoivat omien ohjelmien toteuttamisessa, sillä niiden avulla voitiin todeta, miten ohjelmat toimivat. Myös valmiiden, avoimeen lähdekoodiin perustuvien ohjelmien muokkaamista omiin tarpeisiin pidettiin vaihtoehtona.

Drupalin Whereis-moduulin²⁴ avulla on mahdollista lähettää GPS-vastaanottimella varustellun puhelimen hakemat koordinaatit suoraan Drupaliin. Moduuli sisältää Drupalin päähän asennettavien tiedostojen lisäksi S60-puhelimille tarkoitettua PyS60-kielellä tehdyn ohjelman. Ohjelman käyttöä varten täytyy puhelimeen asentaa Python for S60 ja PyS60 Script Shell²⁵. Ohjelma itsessään on yksinkertainen, sillä se vain käynnistää puhelimen GPS:n ja välittää sen hakemat koordinaatit ohjelmaan. Kun koordinaatit on haettu, voi datan lähettää Drupaliin ohjelman valikoi-

²¹<http://wiki.forum.nokia.com>

²²http://wiki.forum.nokia.com/index.php/Portal:Programming_Languages

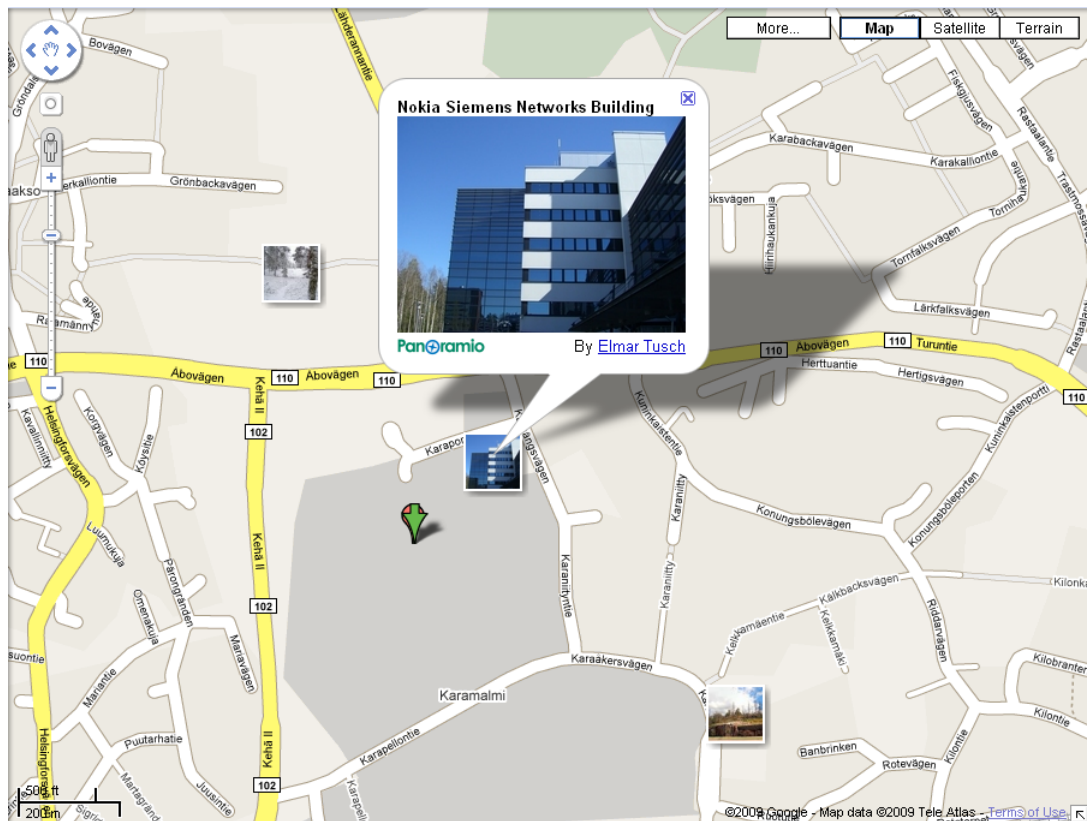
²³http://www.forum.nokia.com/Technology_Topics/Device_Platforms/S60/Platform_Security/Capability_Descriptions.shtml

²⁴<http://drupal.org/project/whereis>

²⁵<http://sourceforge.net/projects/pys60>

den kautta XML-RPC-rajapintaa²⁶ käyttäen. Moduuli mahdollistaa myös tunnuk-sien käyttämisen, joten koordinaattien lähettäminen voidaan yksilöidä Drupalissa, jolloin tiedetään, kenen lähettämä mikäkin data on.

Paikkatiedon hakemista ja lähettämistä varten moduuli tuntui aluksi soveltuvan hyvin, sillä tarvittavien ohjelmien asennus ei ollut vaikeata. Aikaa se kuitenkin vie, mikä asettaa heti kynnyksen moduulin käytölle UCL:n käyttäjiä ajatellen. Lisäk-si koordinaattien hakeminen toimii GPS:n ehdoilla, joten sisätiloissa koordinaattien saaminen kestää ensimmäisellä käynnistyskerralla noin minuutin. Seuraavilla käyttö-kerroilla, ennen puhelimen uudelleenkäynnistystä, koordinaatit saadaan ohjelmaan muutamissa sekunneissa. Ulkona moduulin käyttö on GPS:n tapaan selvästi no-peampaa. Ohjelmaa testattiin Nokia 5800 XpressMusicin sisäisen GPS:n ja Nokia N73 -puhelimeen liitetyn ulkoisen GPS:n kanssa. Ensiksi mainitun puhelimen käyttö-esimerkkitulo näkyy kartalla kuvassa 5.3. Moduuliin suunniteltiin liitettäväksi myös puhelimen kameralla otettava kuva, joka rikastaisi paikkatietoa entisestään. UCL-verkkoympäristöön moduulia ei kuitenkaan kelpuutettu, sillä käyttäjiltä vaaditta-vien asennuksien lisäksi moduulin käyttö rajautuu S60-puhelimiin, joissa on oltava GPS-vastaanotin. Moduuli ei ole myöskään päivittynyt sitten julkaisun (13.6.2009), joten lisätoiminnot täytyisi toteuttaa siihen itse.



Kuva 5.3: Whereis-moduulin paikannustulos Goole Mapsin kartalla

²⁶<http://www.xmlrpc.com>

määrästä johtuen ei toteutukselle löytynyt yhtä ainuttakaan vaihtoehtoa, jolla olisi voitu tunnistaa ja saada automaattisesti kaikista päätelaitteista tarvittavat tiedot. Niinpä paikannuksen tapaan tunnistusvaihtoehtoja tarvittiin useampi.

5.4.1 Teeman valinta

Kuten luvussa 5.1 kerrotaan, UCL-ympäristössä laite tunnistetaan ensin Mobile Pluginin²⁸ avulla. Moduulin käyttämä WURFL tunnistaa laitteen tyyppin UCL:n Drupal-asennukseen määriteltyjen sääntöjen mukaisesti. Säännöt ovat listauksessa 5.3. Kun WURFL palauttaa työpöytäselainta käytettäessä *is_wireless_device: false*, käytetään oletusteemaa ja sen mukaista tyyli tiedostoa, joka ei näy listauksessa 5.3. Mobiililaitteen tapauksessa ensimmäisenä tutkitaan, tunnistiko WURFL mobiiliselaimeksi Nokian. Toisena tutkitaan, oliko USERAGENT-tiedoissa Opera Mini ja kolmentena, oliko kyseessä langaton laite eli tässä mobiililaitte. Näiden perusteella valitaan yksi listauksen 5.3 oikeassa laidassa olevista tyyli tiedostoista eli joko Nokia, Opera Mini tai muu mobiililaitte. Näiden lisäksi WURFL:lla tunnistetaan, onko päätelaitteessa kosketusnäyttö. Kosketusnäytön tapauksessa WURFL palauttaa *pointing_method: touchscreen*, jolloin mobiilityyli tiedostoissa otetaan luokka *.touchscreen* käyttöön. Kosketusnäyttöpuhelimille kasvatetaan kuvakkeiden, linkkien ja tekstien kokoa sekä lisätään sivuston oikeaan laitaan tyhjää tilaa selaamiselle, jotta UCL:n käyttö onnistuisi sujuvammin.

Listaus 5.3: Selaintunnistukseen käytetyt säännöt, joilla valitaan oikea tyyli tiedosto

```

1 mobile_browser =~ Nokia ? group=nokiabasic
2 USERAGENT =~ Opera Mini ? group=operamini
3 is_wireless_device == true ? group=mobile

```

5.4.2 Laitetietojen kerääminen käyttökokemuslomakkeelle

Laitetietojen keräämiseen käyttökokemuslomakkeelle ei yksinään Mobile Plugin -moduuli riitä, koska sillä ei saada työpöytäselainta tai Opera Miniä käyttävästä laitteesta kaikkia tarvittavia tietoja. Apuna työpöytäselaimille käytetään Drupalin Browscap -moduulia²⁹, jonka käyttöä ja tuloksia esitellään luvussa 3.6.2. Moduulin avulla haetaan ensin selain, minkä jälkeen haetaan Mobile Plugin -moduulilla laitetiedot sekä `$_SERVER`-muuttujalla laitteen IP-osoite listauksen 5.4 mukaisesti. Roshan Bhattarain (2007) kirjoittaman PHP-funktion mukaisesti IP-osoitteen saamiseksi käytetään kolmea eri ympäristömuuttujaa, joita käytetään listauksen 5.4 mukaisessa järjestyksessä. Jos edellinen palauttaa arvoksi tyhjän, yritetään seuraavaa

²⁸<http://drupal.org/project/mobileplugin>

²⁹<http://drupal.org/project/browscap>

ympäristömuuttujaa. Jos kaikki kolme palauttavat tyhjän arvon, merkataan IP:ksi tyhjä arvo, eikä tällöin suoriteta IP-paikannusta. Kun IP-osoite jollain näistä löytyy, sijoitetaan se lomakkeelle ja suoritetaan IP-paikannus hostip.infon API:n³⁰ avulla. Tulokseksi saadusta XML-dokumentista jäsennetään lomakkeelle kaupunki, maa ja koordinaatit.

Listaus 5.4: Jokaisesta laitteesta haetaan perustiedot eri menetelmin

```

1 $browser = browscap_get_browser(null, true);
2 $device = mobileplugin_get_wurfl_device();
3
4 function getRealIpAddr()
5 {
6     if (!empty($_SERVER['HTTP_CLIENT_IP']))
7     {
8         //check ip from share internet
9         $ip=$_SERVER['HTTP_CLIENT_IP'];
10    }
11    elseif (!empty($_SERVER['HTTP_X_FORWARDED_FOR']))
12    {
13        //to check ip is pass from proxy
14        $ip=$_SERVER['HTTP_X_FORWARDED_FOR'];
15    }
16    else
17    {
18        $ip=$_SERVER['REMOTE_ADDR'];
19    }
20    return $ip;
21 }
22 //Sijoitetaan IP-osoite lomakkeelle.
23 $form['field_ip_address'][0]['#default_value'] =
24 array('value' => getRealIpAddr());

```

Seuraavaksi tutkitaan Browscapin avulla, onko käyttäjän laite mobiili vai ei eli palauttaako moduuli arvoksi true vai false. Jos tulokseksi saadaan, että laite ei ole mobiili, täytetään lomakkeen käyttöjärjestelmä-, selain- ja selainversiokenttä browscap.ini-tiedostosta³¹ listauksen 5.5 mukaisesti. Näiden lisäksi sijoitetaan näytön pysty- ja vaakaresoluutio lomakkeelle joko sovelluskirjanmerkin tiedoista tai jos sovelluskirjanmerkkiä ei käytetä, luetaan resoluutio JavaScriptillä.

³⁰<http://api.hostip.info>

³¹<http://browsers.garykeith.com>

Listaus 5.5: Lomakkeelle haetaan työpöytäselaimen tapauksessa tiedot Browscapin avulla

```

1 //Sijoitetaan lomakkeelle browscap.ini:llä saatavat tiedot.
2 $form['group_technical']['field_os']['value']
3 ['#value'] = $browser['platform'];
4 $form['group_technical']['field_browser']['value']
5 ['#value'] = $browser['browser'];
6 $form['field_browser_version'][0]['#default_value'] =
7 array('value' => $browser['version']);
8
9 //Sijoitetaan lomakkeelle resoluutio joko sovelluskirjan-
10 //merkillä tai lukemalla tiedot suoraan JavaScriptillä.
11 if ($_GET["width"] != NULL) {
12 $form['field_resolution_width'][0]['#default_value'] =
13 array('value' => $_GET["width"]);
14 $form['field_resolution_height'][0]['#default_value'] =
15 array('value' => $_GET["height"]);
16 } else {
17 $getscreen = '
18 $(document).ready(function() {
19   $("#input#edit-field-resolution-width-0-value").
20   val(screen.width);
21   $("#input#edit-field-resolution-height-0-value").
22   val(screen.height);
23 });';
24 }

```

Näillä keinoilla saadaan siis työpöytäselaimen tapauksessa käyttöjärjestelmä, selain ja sen versio, IP-osoite sekä resoluutio käyttökokemuslomakkeelle. Ainoastaan päätelaitteen ja käytetyn yhteyden tyyppi jäävät käyttäjälle täytettäväksi, sillä tavanomaisia selaimia, kuten Firefoxia, Operaa ja Safaria on mahdollista käyttää pöytäkoneella, kannettavalla tai miniläppärillä. Myöskään yhteystyyppin erotteluun ei löydetty ratkaisua.

Matkapuhelin tapauksessa laitetiedot piilotetaan näkyvistä, sillä tällöin saadaan kerättyä tarvittavat tiedot automaattisesti. Tämä mahdollistaa myös sen, että lomake on kevyempi avata mobiililaitteilla, sillä ladattavaa sisältöä on vähemmän. Tietojen piilottaminen mahdollistaa myös sen, että teknisten tietojen oletukset eli niin sanottu arvaus saadaan talteen sellaisenaan. Käyttäjä voi myöhemmin muokata tallennettuja laitetietoja työpöytäselaimella tai matkapuhelimella työpöytänäkyvässä. Mobiililaitteista ensimmäisenä haetaan Opera Minin otsikoiden avulla laitteen merkki ja malli. Kuten on listauksessa 5.6, seuraavaksi haetaan selaintiedot samaan tapaan kuin työpöytäselaimesta. Lopuksi haetaan vielä WURFL:lla pääte-

laitteen merkki, malli, selain ja sen versio sekä näytön resoluutio. Näillä tiedoilla korvataan mahdolliset Opera Minin otsikoiden avulla saadut tiedot, koska oletetaan että WURFL:n saadaan tiedot vain, kun selaimena ei ole Opera Mini.

Listaus 5.6: Menetelmät joilla mobiililaitteen tiedot kerätään

```

1 //Piilotetaan tekniset tiedot mobiililaitteen tapauksessa.
2 $form['group_technical']['#attributes'] =
3 array('class' => 'group-technical hidden');
4
5 //Luetaan ensimmäisenä Opera Minin päätelaitetiedot.
6 $phone = explode('#', $_SERVER['HTTP_X_OPERAMINI_PHONE']);
7 //Erotellaan puhelimen merkki ja malli #-erotinmerkin avulla
8 //(esim. Nokia # N73).
9 $brand = trim($phone[0]);
10 $model = trim($phone[1]);
11
12 //Selain saadaan työpöytäselaimen tapaan Browscapilla.
13 $form['group_technical']['field_browser']['value']
14 ['#value'] = $browser['browser'];
15
16 //Haetaan laitteen merkki ja malli Mobile Pluginilla.
17 $brand = $device->getCapability('brand_name');
18 $model = $device->getCapability('model_name');
19 $form['group_technical']['field_mobile_brand']['value']
20 ['#value'] = $brand;
21 $form['group_technical']['field_mobile_model']['value']
22 ['#value'] = $model;
23
24 //Jos päätelaite on mobiililaite, oletetaan että
25 //laitetyyppi on matkapuhelin ja yhteys on matkapuhelinverkko
26 $form['field_connection']['#default_value'][0] =
27 array('value' => 'mobile');
28 $form['field_device_type']['#default_value'][0] =
29 array('value' => 'Mobile phone');
30
31 //Haetaan päätelaitetiedot WURFL:lla.
32 $form['group_technical']['field_browser']['value']
33 ['#value'] = $device->getCapability('mobile_browser');
34 $form['field_browser_version'][0]['#default_value'] =
35 array('value'=>$device->getCapability
36 ('mobile_browser_version'));
37 $form['field_resolution_width'][0]['#default_value'] =

```

```

38 array ( 'value'=>$device->getCapability ( 'resolution_width '));
39 $form [ 'field_resolution_height '][0][ '# default_value ' ] =
40 array ( 'value'=>$device->getCapability ( 'resolution_height '));

```

Näiden lisäksi myös Opera Minin demo³² on otettu huomioon, sillä sen käyttöä pidettiin mahdollisena opiskelijoiden viikkoharjoitustehtävissä. Opera Minin demo palauttaa edellä esitetyillä Opera Minin otsikkotiedoilla päätelaitteen merkiksi ja malliksi kysymysmerkin. Tässä tapauksessa tehdään poikkeus ja haetaan tiedot WURFL:n avulla, jolloin saadaan kysymysmerkkien sijaan lomakkeelle päätelaitteen merkiksi Opera ja malliksi Mini. Muut tiedot haetaan edellä esitetyin keinoin. Tietojen hakeminen demon tapauksessa on esitetty listauksessa 5.7, josta nähdään, että jos merkki tai malli ei ollut kysymysmerkki, sijoitetaan lomakkeelle aiemmin haetut tiedot.

Listaus 5.7: Opera Minin demon tiedot haetaan WURFL:lla

```

1  if ($brand == '?') {
2  $form [ 'group_technical '][ 'field_mobile_brand '][ 'value ' ]
3  [ '# value ' ] = $device->getCapability ( 'brand_name ');
4  } else {
5  $form [ 'group_technical '][ 'field_mobile_brand '][ 'value ' ]
6  [ '# value ' ] = $brand;
7  }
8  if ($model == '?') {
9  $form [ 'group_technical '][ 'field_mobile_model '][ 'value ' ]
10 [ '# value ' ] = $device->getCapability ( 'model_name ');
11 } else {
12 $form [ 'group_technical '][ 'field_mobile_model '][ 'value ' ]
13 [ '# value ' ] = $model;
14 }

```

5.4.3 Laitetunnistuksen tuloksia

Automaattisen laitetunnistuksen tuloksien arviointiin pohdittiin muutamia eri keinoja. Yksinkertaisin tapa on selvittää UCL:n käyttämän palvelimen lokitietojen avulla, mitä tietoja käyttökokemuksiin on tallennettu ja mitä tietoja on jälkepäin muutettu. Tämä onnistuu Drupalissa kätevästi asettamalla versiointi päälle käyttökokemuslomakkeiden tallentamisen yhteyteen. Kun käyttökokemusta muokataan ja muokkaukset tallennetaan, versioinnin avulla voidaan vertailla kahta eri tallennusta

³²<http://www.opera.com/mobile/demo/?ver=4>

ja siten nähdä, mitkä tiedot lomakkeella ovat muuttuneet. Versiointi Drupalissa toimii vastaavasti kuin se toimii esimerkiksi Wikipediassa³³. Toinen vaihtoehto olisi kysyä käyttäjiltä, miten heidän laitteensa tunnistettiin ja joutuivatko he muuttamaan automaattisesti saatuja tietoja. Samaa kyselyä harkittiin kolmantena vaihtoehtona myös UCL-työryhmälle, joka testasi laitetunnistusta monilla eri laitteilla. Tuloksien arvioinnissa päädyttiin versioinnin avulla suoritettavaan tarkasteluun ja myös omiin testauksiin luvussa 5.5 esiteltyjen testiympäristöjen ansiosta. Versiointia ei kuitenkaan saatu toimimaan aivan niin kuin haluttiin. Nykyisessä toteutuksessa automaattisten laitetietojen haun tuloksia ei saada talteen ennen kuin käyttäjä tallentaa käyttökokemuslomakkeen. Tämä tarkoittaa, että käyttäjä saattaa muokata laitetietoja ennen tallentamista ja siten poistaa alkuperäisen tiedon lomakkeelta. Tämän niin sanotun arvauksen talteen saaminen koettiin teknisesti liian työlääksi. Toisaalta tämä on ongelma ainoastaan työpöytäselaimen näkymässä, jossa laitetiedot näkyvät käyttäjälle aina, kun taas mobiilinäkymässä laitetiedot ovat piilotettuina.

Testiympäristöjen avulla saatuja tuloksia eri päätelaitteilla on listattu liitteessä 1. Liitteeseen on ensin listattu mobiililaitteet ja sitten tavanomaiset selaimet työpöytäkoneissa ja kannettavissa, joilla UCL:ää on testattu. Laitteet ja selaimet on valikoitu mobiililaitteiden osalta satunnaisesti testiympäristössä vapaana olleista laitteista. Mahdollisimman monta laitetta on testattu sekä Nokian oman selaimen että Opera Mini 4.2:n kanssa. Työpöytäkoneiden ja kannettavien valinta osui fyysisesti vapaana olleisiin koneisiin siten, että vaihtelua olisi mahdollisimman paljon sekä käyttöjärjestelmän että selaimen osalta. Tuloksista ilmenee, että Opera Miniä käytettäessä saadaan mobiililaitteista pari poikkeusta lukuun ottamatta kaikki halutut päätelaitetiedot eli päätelaitetyyppi, yhteystyyppi, puhelimen merkki ja malli sekä selain ja sen versio sekä resoluutio. Ainoastaan viimeinen näistä on jokaisen kohdalla väärin. Tämä johtuu siitä, että Opera Mini *renderöi* (engl. rendering) eli muuttaa sisällön asetelua laitteille sopivaksi. Leveydeksi asetetaan tällöin 128 pikseliä ja korkeudeksi 96 pikseliä, joten periaatteessa UCL:n käyttökokemuslomakkeelle saatu resoluutio ei ole väärä. JavaScriptillä saataisiin myös todellinen näytön resoluutio komentoilla `window.innerWidth` ja `window.innerHeight`, mutta tätä ei UCL:ssä ehditty kokeilemaan. (vrt. Storey, 2007) Poikkeuksen muissa laitetiedoissa tekevät Nokia N96, Nokia N97 ja Nokia 6700 slide, joista kaksi viimeistä eivät olleet vielä testausvaiheessa ilmestyneet markkinoille myyntiin. Yhdestäkään näistä ei saada puhelimen mallia, jonka kenttä jää Opera Minin HTTP-otsikoita käytettäessä tyhjäksi. Syytä tähän on varsinkin Nokia N96:n tapauksessa vaikea sanoa, sillä puhelin oli testausvaiheessa ollut markkinoilla jo noin kaksi vuotta. Muiden laitteiden osalta tiedot ovat oikein käytettäessä Opera Miniä.

³³<http://wikipedia.org>

Nokian oman selaimen osalta tilanne on hieman parempi, sillä WURFL:lla³⁴ saadaan selaimen versiota lukuun ottamatta kaikki tiedot oikein riippuen siitä, onko WURFL-tiedot päivitetty ajan tasalle verkkoympäristöön. Nokian selaimen versiota ei jostain syystä ole WURFL-tietoihin laitettu yhdenkään Nokian puhelimen osalta, vaikka UAProf-tiedoista versio löytyy esimerkiksi Nokia N96:n³⁵ osalta, kuten nähdään listauksessa 5.8. Kyseisen puhelimen selain on Nokia ja versio 4.0. WURFL-tiedoissa³⁶ tämän puhelimen selaimena on kuitenkin Safari. Samoin kuin on Nokia E70:n WURFL-tiedoissa³⁷. Nokia E70:n tapauksessa tilanteesta tekee mielenkiintoisen se, että Nokia tarjoamissa UAProf-tiedoissa³⁸ selaimen kuvauksessa on Mozilla ja versiona 5.0, mikä näkyy listauksessa 5.9. Niinpä tässä tullaan tulokseen, jossa kummastakin tietokannasta löytyy virhe, joka johtaa väärin laitetietojen tunnistukseen.

Listaus 5.8: Nokia N96:n selaimen kuvaus RDF-dokumentissa

```

1 <prf:component>
2 <rdf:Description rdf:ID="BrowserUA">
3 <rdf:type rdf:resource="http://www.openmobilealliance.org/
4 tech/profiles/UAPROF/ccppschem-20021212#BrowserUA"/>
5 <prf:BrowserName>Nokia</prf:BrowserName>
6 <prf:BrowserVersion>4.0</prf:BrowserVersion>
7 ...
8 </rdf:Description>
9 </prf:component>

```

Listaus 5.9: Nokia E70:n selaimen kuvaus RDF-dokumentissa

```

1 <prf:component>
2 <rdf:Description rdf:ID="BrowserUA">
3 <rdf:type rdf:resource="http://www.openmobilealliance.org/
4 tech/profiles/UAPROF/ccppschem-20021212#BrowserUA"/>
5 <prf:BrowserName>Mozilla</prf:BrowserName>
6 <prf:BrowserVersion>5.0</prf:BrowserVersion>
7 </rdf:Description>
8 </prf:component>

```

Toisin kuin Opera Minillä, WURFL:n avulla saadaan resoluutio oikein. Nokia 6700 sliden resoluutio on väärä, sillä vaikka se löytyy WURFL-tiedoista³⁹, niin

³⁴<http://wurfl.sourceforge.net>

³⁵<http://nds1.nds.nokia.com/uaprof/NN96-1r100.xml>

³⁶<http://db.wurflpro.com/device/show/22882>

³⁷<http://db.wurflpro.com/device/show/15336>

³⁸<http://nds1.nds.nokia.com/uaprof/NE70-1r100.xml>

³⁹<http://db.wurflpro.com/device/show/26750>

UCL:n WURFL-tiedot eivät olleet testausvaiheessa ajan tasalla. Siksi myös puhelinmallia ei tunnistuksessa saatu lomakkeelle. Sama koskee myös Nokian puhelinta N97, joka ilmestyi markkinoille loppusyksystä 2009. Vanhempien puhelimien osalta tunnistus toimii hyvin, edellä listattuja poikkeuksia lukuun ottamatta.

Myöskään työpöytäselainten osalta laitetiedot eivät aina ole aina täydelliset. Firefoxia käytettäessä saadaan automaattisesti Browscapin avulla käyttöjärjestelmä, selain ja sen versio. Lisäksi listauksen 5.4 mukaisesti saadaan päätelaitteesta riippumatta sen IP-osoite, jolla voidaan tarvittaessa tehdä IP-paikannus, jos selain ei tue Geolocation API:a. Myös Internet Explorerista, Chromesta, Safarista ja Operasta saadaan samat tiedot, mikäli ne löytyvät Browscap-moduulin browscap.ini-tiedostosta. Jos tiedostoa ei ole päivitetty, eikä siten selain sieltä löydy, lomakkeelle ilmestyy selaimen kohdalle "Default Browser", jonka versio on 0. Firefoxista ja Google Chrome 5.0 betasta saadaan luettua resoluutio, mutta muista työpöytäselaimista ei tätä tietoa saada. Ilmeisesti JavaScript-koodia tulisi muuttaa jokaisen selaimen osalta, jotta resoluutio saataisiin talteen. Tämän lisäksi ongelmia aiheuttaa Windows XP Professional x64 Edition - käyttöjärjestelmä, sillä Browscap tunnistaa tämän Win2003:ksi, joka on palvelimille tarkoitettu käyttöjärjestelmä. Syy tähän on, että sekä Windows 2003 x64 että XP x64 käyttävät identtistä kerneliä eli käyttöjärjestelmän ydintä (Wikipedia, 2010). Tämä ei sinänsä ole väärä tieto, sillä myös Windowsin omat käyttöjärjestelmätiedot ilmoittavat molempien Windowsien versioksi 2003. Testin ainoalla Linux-koneella testatessa selaintiedot saadaan lomakkeelle oikein Firefoxilla, mutta Ubuntu 8.04:n sijaan käyttöjärjestelmäksi merkataan vain Linux. Tämän tiedon tarkentaminen jää siis käyttäjän muokattavaksi.

Käyttökokemusten versioinneista ilmeni, että samat ylläesitetyt ongelmat toistuvat myös muiden käyttäjien käyttökokemuksissa. Niinpä Opera Minillä täytetyt lomakkeet ovat teknisten tietojen osalta resoluutiota lukuun ottamatta oikein. Samoin Firefoxilla saatiin hyviä tuloksia, joissa tekniset tiedot näyttivät olevan oikein, vaikkei siis versiointia näistä saatukaan. Vain pari käyttäjä teki käyttökokemuksensa mobiililaitteilla, joten päätelaitetunnistuksen onnistuneisuutta on vaikea arvioida. Versiointien perusteella muutoksia päätelaitetietoihin ei mobiililaitteiden tapauksessa ole tehty. Sen sijaan osa opiskelijoista kirjasi käyttökokemuksensa eri laitteella kuin sillä, jolla todellisuudessa käytti arvioitavaa verkkopalvelua. Niinpä teknisiin tietoihin tallentui yhdistelmiä eri päätelaitteista. Esimerkiksi muutamissa työpöytäselaimella kirjatuissa käyttökokemuksissa esiintyi päätelaitteena Nokian puhelin, mutta selain olikin Firefox ja resoluutio oli 22-tuumaisen näytön resoluutio.

5.5 Testiympäristöt

Projektin alkuvaiheista lähtien tiedettiin olevan tarvetta testilaitteille, erityisesti matkapuhelimille ja GPS-vastaanottimille. Rajoitteena laitteiden haalimiselle oli,

että niiden käytön tulisi olla ilmaista, mutta toisaalta eri testilaitteita tarvittaisiin mahdollisimman monta. Testaamiseen oli fyysisesti saatavilla Nokian N73 -puhelin ja Soneran mobiililaajakaista eli mokka. Muita laitteita ei työpöytäkooneita ja kannettavia lukuun ottamatta ollut saatavilla, joten testiympäristöt olivat yksi keskeisimmistä osioista UCL-verkkoympäristön toteutusvaiheessa. Toteutuksen toimivuutta haluttiin testata mahdollisimman monella laitteella ja säätää UCL:n ulkoasua niiden mukaisesti. Testausmahdollisuuksien kartoitus oli siinäkin mielessä oleellista, että kehittäjien ja muiden aiheesta kiinnostuneiden on hyvä tietää, mitä ilmaisia vaihtoehtoja testaamiselle on. Tässä luvussa on esitelty UCL:n suunnittelu- ja toteutusvaiheessa kaikkein käytetyimmät testiympäristöt, jotka ovat kaikkien saatavilla testaamista varten.

5.5.1 Remote Device Access Services

Forum Nokia⁴⁰ tarjoaa matkapuhelinkehittäjille mahdollisuutta käyttää Nokian puhelimia Internetin välityksellä. Palveluita on kaksi; RDA eli Remote Device Access⁴¹ ja VDL eli Virtual Developer Lab⁴². Nokia kutsuu näitä nimellä Remote Device Access Services⁴³. Näistä VDL on maksullinen palvelu, joka tarjoaa 1500 matkapuhelinta yli 25 eri operaattorin liittymällä ympäri maailman. Palvelu on tehty mobiilikkehittäjille, jotta nämä voisivat tuoda markkinoille entistä nopeammin sovelluksia ja muuta sisältöä. Palvelua voi käyttää kolmen tunnin koeajan ilmaiseksi. Tämän jälkeen käyttö on maksullista. (Forum Nokia, s.a.-c) UCL:n testaamiseen palvelua ei käytetty sen maksullisuuden takia. Lisäksi Forum Nokian RDA:n tarjoamat puhelimet riittivät hyvin testaustarpeisiin.

RDA on ilmainen palvelu, jossa voi Forum Nokian tunnuksilla käyttää yli 40 eri Nokian matkapuhelinmallia. Myös tunnusten rekisteröinti on maksutonta. Palvelun avulla Nokia on halunnut antaa sovelluskehittäjille mahdollisuuden testata omia ohjelmiaan milloin ja missä vain. Kuvassa 5.5 näkyy RDA:n toimintaperiaate. Käyttäjät ottavat selaimella yhteyden RDA:n palvelimelle Internetin yli eli mistä päin maailmaa tahansa. Puhelimet näkyvät kuvan 5.6 mukaisesti sivustolla listauksena. Puhelimia voi varata yhden vuorokauden aikana korkeintaan kahdeksaksi tunniksi, vähintään 15 minuutin aikajaksoissa. Varauksia voi myös tehdä useita päiviä etukäteen. Kun puhelin on varattu, eivät muut voi käyttää sitä. Puhelimet ovat yhteydessä RDA:n palvelimeen, joka välittää puhelimen näyttöjen kuvat Internetin yli käyttäjille, kun nämä haluavat yhteyden puhelimeen. Näytön lisäksi RDA:n ikkunassa, kuvassa 5.7, ovat muun muassa puhelimen näppäimet (myös sivunäppäimet kuten

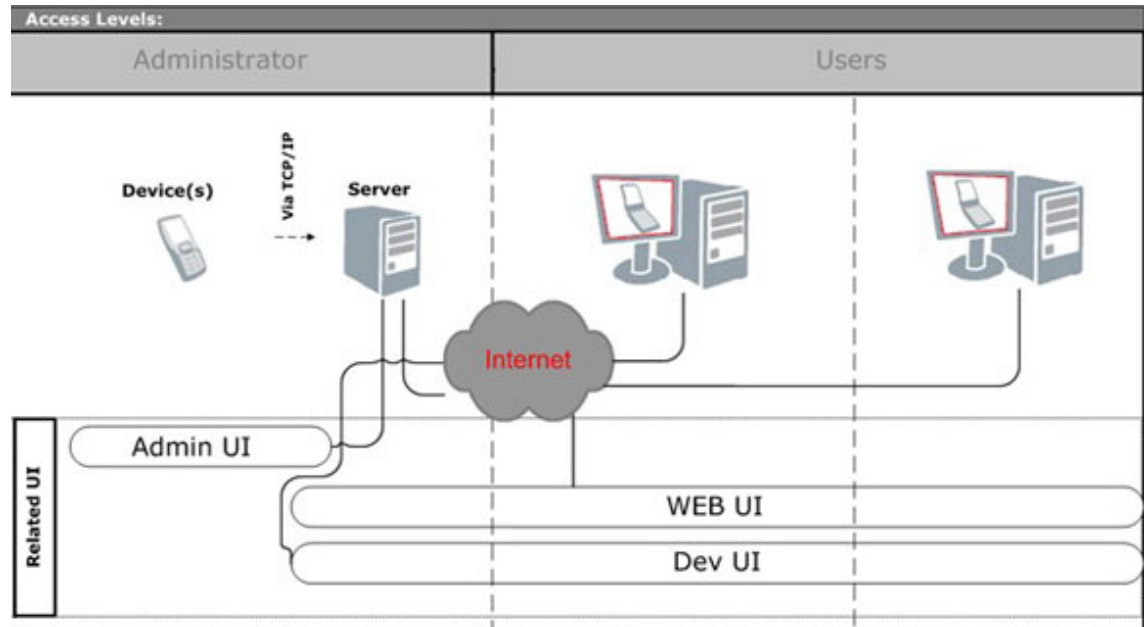
⁴⁰<http://www.forum.nokia.com>

⁴¹<http://apu.ndhub.net>

⁴²<http://www.deviceanywhere.com>

⁴³<http://www.forum.nokia.com/rda>

kameran käynnistin), pikanäppäimet kuvankaappaukselle ja videoinnille sekä lokille, josta näkee puhelimen antamat virheilmoitukset. Samaan aikaan on mahdollista käyttää useita puhelimia, mikä mahdollistaa esimerkiksi Bluetoothin testaamisen kahden puhelimen välillä. (Forum Nokia Wiki, 2009d)



Kuva 5.5: RDA:n toimintaperiaate (Forum Nokia Wiki, 2009d)



Phones available to you

> Change your local time: 18 Mar 15:56 > List all phones (reset filter)

- Filter by tag -

Move your mouse on top of a phone, choose reservation length and click Start. A Java application will be downloaded to your computer for using the phone.

You can make about 8 hours worth of reservations today

Available now

<p>Nokia 5320 XpressMusic 354826020021572 +358504877263 SW 5.16</p>	<p>Nokia 5500 Sport 351855010132985 +358504877445 SW 4.60</p>	<p>Nokia 5530 XpressMusic 359368036340766 +358447835592 SW 20.0.080</p>	<p>Nokia 5630 XpressMusic 004401104982661 +358504877463 SW 12.020</p>
<p>Nokia 5730 XpressMusic 351964030018248 +358447835663 SW 200.12.80</p>	<p>Nokia 6110 Navigator 356416010444907 +358504858039 SW 6.01</p> <p>Reserve for 30 minutes</p> <p>Available for 18 minutes</p> <p>Start</p> <p>Make later reservation</p> <p>+358504858033 SW 3.0633.09.04</p>	<p>Nokia 6120 classic 356255010014199 +358504858041 SW 7.10</p>	<p>Nokia 6220 classic 352921020020674 +358504877454 SW 5.15</p>
<p>Nokia 6710 Navigator 004401107215770 +358504877451 SW 22.013</p>	<p>Nokia E55 355217030010376 +358504864234 SW 22.009</p>	<p>Nokia E61i 351879010031427 no SIM SW 3.0633.69.00</p>	
<p>Nokia E66 352942020103922 +358504803320 SW 300.21.012</p>	<p>Nokia E71x 004401102970270 no SIM SW 3.28 (AT&T variant)</p>	<p>Nokia E75 359557010007850 +358504877461 SW 202.12.01</p>	

Kuva 5.6: RDA:n selainnäkömä käyttäjille



Kuva 5.7: RDA:n käyttöliittymä - puhelimenä Nokia N73

Palvelu vaatii toimiakseen käyttäjätunnuksen lisäksi JRE (Java Runtime Environment) version 5 tai uudemman ja selaimen, joka tukee JavaScriptia sekä Internet-yhteyden, jonka portit 80 ja 1200 on oltava auki ulospäin RDA-palvelimelle. RDA:n puhelimilla voi testata Symbian-, Java- ja Adobe Flash Lite -sovelluksia. Myös Python-, Open C- ja Qt-sovelluksia on mahdollista testata, kunhan ohjelmien ajoon tarvittavat lisäosat asentaa ensin puhelimiin. Sovellusten lisäksi voi testata puhelimen teemoja, valikoita, valmiita ohjelmia, selata Internetiä ja käynnistää puhelimen uudestaan. Puhelimeissa on Elisän SIM-kortti, jonka avulla voi käyttää 2G- ja 3G-verkkoa. WLAN-puhelimilla voi ottaa yhteyden RDA:n tarjoamaan tukiasemaan, mikä nopeuttaa selaamista ja sisällön lataamista puhelimiin. Testailua varten puhelimiin voi soittaa ja lähettää tekstiviestejä, mutta niillä soittaminen ja viestien lähettäminen ei onnistu. Osassa puhelimista on GPS-vastaanotin, joka tosin toimii heikosti siksi, että laitteet sijaitsevat sisätiloissa. Paikkatietoisten palveluiden testaamista varten on kehitetty työkalu nimeltään Simulationpsy, joka simuloi puhelimeen GPS-koordinaatit niitä pyydettyäessä. Puhelimiin voi siirtää omia tiedostoja palvelun tiedostonhallintaohjelman avulla. Palvelu on ilmainen, eikä Nokia takaa, että kaikkiin puhelimiin on aina pääsy. Myös katkoja palvelussa voi ilmetä. Eikä tek-

nistä tukea anneta, mutta Forum Nokian Wikisivuilta saa tarvittaessa apua muilta käyttäjiltä. (Forum Nokia, s.a.-b; Forum Nokia, 2009)

RDA:ssa on puhelinmalleja siis kaikkiaan yli 40 ja suosituimpia puhelimia on useampi kappale käytettävissä, mikä nostaa puhelimien kokonaismäärän noin 60:een. Tarjolla on enimmäkseen vain uusimpia malleja, joissa on alustana S60 3rd Edition tai uudempi. Niinpä esimerkiksi Series 40 -puhelimia ei ole, vaan halutessaan niitä täytyy pyytää väliaikaisesti saataville. Sen sijaan maksullisesta VDL:sta löytyy vanhempia malleja. RDA:ssa on eri resoluutioita useita, mikä helpottaa erityisesti verkkosivujen asettelun testaamista eri näytöillä. (Forum Nokia Wiki, 2010) RDA:n puhelimet toimivat siis etäyhteyden kautta aivan kuten tavallisetkin puhelimet, soittamista ja viestien lähettämistä lukuun ottamatta. UCL:n testaamiseen palvelua käytettiin todella paljon, sillä verkkoympäristön mobiiliteemoja haluttiin testata mahdollisimman monella laitteella. Lisäksi laitetietojen automaattiseen tunnistamiseen palvelu sopi erinomaisesti, koska puhelimia on paljon. Myös Opera Minin asentaminen ja testaaminen onnistuivat puhelimilla, joten RDA sopi siihenkin hyvin. Näiden lisäksi omien ohjelmien asennusta ja ajamista kokeiltiin, jotta saatiin selville, onko paikkatietoa mahdollista saada puhelimista ja asentuuiko omat ohjelmat ylipäätänsä puhelimiin. RDA:n avulla säästettiin siis huomattavasti resursseja, sillä laitteiden ei tarvinnut olla fyysisesti läsnä, eikä niitä tarvinnut hankkia mistään. Tosin on huomattava, että etäpuhelimilla käyttökokemus jää puutteelliseksi, sillä palvelua on käytettävä selaimen kautta ja näppäimistön sekä hiiren avulla on selvästi helpompi selaila Internetiä ja täyttää esimerkiksi lomakkeita. Tämän ja laitteen liikuteltavuuden takia UCL:n testeissä käytettiin myös paria puhelinta, jotka olivat fyysisesti saatavilla testaukseen.

5.5.2 Opera Mini -simulaattori

Verkkosivuston ulkoasun nopeaan tarkasteluun mobiililaitteita varten sopii parhaiten Opera Minin ilmainen simulaattori⁴⁴ versiosta 4 (uudempaa versiota⁴⁵ ei UCL:ssä testattu), joka on kaikkien käytettävissä. Sen tarkoitus on toimia kuten matkapuhelimessa ja täten helpottaa selaimen käyttöönottoa. Samalla se antaa mahdollisuuden testata verkkosivuja, sillä kyseisellä Java-simulaattorilla pääsee selailmaan Internetiä. Selailun lisäksi muita toimintoja ei ole, eikä esimerkiksi näytönkoko voi vaihtaa. Simulaattori on nopea ja kätevä tapa nähdä, miten verkkosivujen sisältö, kuten tekstin asettelu, kuvat, kirjautuminen ja selaimen tunnistaminen toimivat Opera Minissä. Joitakin eroavaisuuksia simulaattorin ja tavallisen puhelimen Opera Minin välillä on, mutta sen avulla saa kuitenkin hyvän yleiskuvan selaimen

⁴⁴versio 4.2: <http://www.opera.com/mobile/demo/?ver=4>

⁴⁵versio 5: <http://www.opera.com/mobile/demo>

käytöstä ja näkymistä ilman, että tarvitsisi asentaa sitä puhelimeen. Tämä säästää verkkosivun kehittäjän aikaa ja toisaalta auttaa käyttäjiä omaksumaan Opera Minin toiminnot nopeammin.

5.5.3 User Agent Switcher

Laitetunnistukseen on tehty myös selaimen asennettava lisäosa nimeltään User Agent Switcher⁴⁶, joka toimii Firefox-, Flock-, Seanmonkey- ja Songbird-ohjelmissa. Se lisää ohjelman valikkoihin mahdollisuuden vaihtaa selaimen User Agent -profiilia kuten listauksessa 5.10, jossa on ylempänä Firefox-selaimen oletus ja alempana lisäosan avulla vaihdettu Iphone 3.0:n User Agent.

Listaus 5.10: Esimerkki User Agent Switcherin käytöstä

```

1 Firefox 3.6:m User Agent:
2 Mozilla/5.0 (Windows; U; Windows NT 5.2; fi; rv:1.9.2)
3 Gecko/20100115 Firefox/3.6
4
5 Iphone 3.0:n User Agent:
6 Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us)
7 AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341
8 Safari/528.16

```

Lisäosan valikoihin voi luoda uusia UA-tietoja poimimalla puhelimen User Agentin valmistajien sivuilta tai hakemalla niitä hakukoneiden avulla. McQuillan (2009) käyttää esimerkkinä Nokia E71:n laitetietoja⁴⁷, josta voidaan poimia alkuosan kommenteista kyseisen puhelimen User Agent ja tallentaa se lisäosan tietoihin. Kun tämän jälkeen valitaan lisäosan valikoista Nokia E71 ja avataan esimerkiksi BBC:n verkkosivut⁴⁸, huomataan, että sivusto tunnistaa käyttäjän selaimen puhelimeksi. Tällöin sivusto uudelleenohjaa käyttäjän tavanomaisen BBC:n etusivun sijaan mobiilinäkymään⁴⁹. (McQuillan, 2009) Sama menetelmä onnistuu luvussa 3 esitettyjen laitteistotiedonkeruukirjastojen kanssa, joten User Agent Switcherin avulla voi testata mobiililaitetunnistusta pelkällä selaimen User Agentin muutoksella. Lisäosaan voi tallentaa myös lukuisia muita User Agenteja, kuten Internet Explorerin, Operan ja Chromen, jolloin voi testata ilman selaimen vaihtoa, toimiiko laitetunnistuksen lisäksi selaintunnistus oikein. Tämä nopeuttaa ja säästää jälleen testauksen resursseja, sillä laitetunnistuksen ja uudelleenohjauksen toimivuutta voi testata nopeilla sykleillä pelkkää selainta käyttäen. Käyttökokemuksesta jää puuttumaan pieni näytönkoko ja selailu puhelimen näppäimiä käyttäen, mutta lisäosan avulla näkee, miten

⁴⁶<http://chrispederick.com/work/user-agent-switcher>

⁴⁷<http://nds1.nds.nokia.com/uaprof/NE71-1r100.xml>

⁴⁸<http://www.bbc.co.uk>

⁴⁹<http://www.bbc.co.uk/mobile/i/>

sivusto käyttäytyy eri laitteiden kanssa.

5.6 Käyttäjien palaute

UCL-verkkoympäristöä käytettiin heti julkaisun jälkeen kahdella hypermedian kursilla, jotka olivat luvussa 2 mainitut MATHM-47250 Verkkopalvelun sisällöntuotanto ja MATHM-47300 Verkkopalvelun käyttökelpoisuus ja arviointi. UCL:ssä itsenäisesti tehtyjen viikkotehtävien jälkeen hypermedian opiskelijat vastasivat verkkoympäristön käyttöön liittyvään kyselyyn⁵⁰. Kyselyssä käytettiin AttrakDiff-kokemusmittaristoa⁵¹, jota täydennettiin opiskelijoiden vapaamuotoisilla kommentteilla. Kokemusmittariston avulla saatiin tietoa opiskelijoiden suhtautumisen lisäksi siitä, miten voimakkaana opiskelijat kokivat käyttäjäkokemuksen eri ulottuvuuksien laadukkuuden tai sen puuttumisen. Lisäkommenttien avulla saatiin tietoja tekijöistä, jotka vaikuttivat opiskelijoiden suhtautumiseen palautetta antaessa. (Lähteinen & Tervakari, 2010)

Tulosten perusteella UCL-verkkoympäristöön suhtauduttiin positiivisesti ja se koettiin helpoksi, nopeaksi ja yksinkertaiseksi käyttää. Toiminnallisen laadukkuuden jälkeen positiivisimmat kommentit keräsi UCL:n attraktiivisuus, sillä ympäristön ulkoasu koettiin miellyttäväksi ja visuaalisesti houkuttelevaksi. Positiivisesti vaikuttavana tekijänä nähtiin myös UCL:n innovatiivisuus sekä yhteisöllisyys, vaikkakin verkkoympäristössä ei ollut mahdollista kommentoida muiden tekemiä käyttökokemuksia. Negatiivisena puolena nähtiin UCL-verkkoympäristön käyttötarkoituksen jääminen epäselväksi. Lisäksi UCL:n ei katsottu soveltuvan asiantuntijakäyttöön, eikä sen nähty tuovan mitään lisäarvoa muihin arviointityökaluihin verrattuna. Mitään kriittistä ja siten kiireellistä kehittämiskohdetta ei ilmennyt palautteen perusteella, sillä UCL:ään suhtautuminen oli pääosin positiivista. Käyttökokemuksen laadukkuutta voidaan parantaa entisestään selkeyttämällä ohjeistusta siitä, mihin UCL on tarkoitettu ja mitä sen käytöllä voidaan saavuttaa. Opiskelijoille täytyy selventää, että UCL:n ei ole tarkoitus olla asiantuntijoiden käyttöön tarkoitettu arviointityökalu, vaan arviointityötä tukeva apuväline. Sen tehtävä on helpottaa arvioinnin asiantuntijoita erityisesti käyttäjätietojen keräämisessä. Tämän tiedon avulla on helpompi ymmärtää UCL:n käyttökokemuslomakkeen kysymysten rajallisuus ja kysymysten yleisluontoisuus, joilla oli palautteessa negatiivinen vaikutus toiminnalliseen laadukkuuteen. Yhtenä vaihtoehtona käyttäjien suhtautumisen parantamiseen Tervakari ja Lähteinen (2010) esittävät käyttökokemuslomakkeen toteuttamista erikseen asiantuntijakäyttöön ja käyttäjätesteihin. Tällöin UCL:ää voidaan hyödyntää paremmin myös asiantuntija-arvioinneissa. (Lähteinen & Tervakari, 2010)

⁵⁰http://matriisi.ee.tut.fi/hmopetus/kyselyt/ucl/ux_kysely.php

⁵¹<http://www.attrakdiff.de/en/Home>

6. POHDINTAA

UCL:n päätelaitetunnistukseen ja käyttäjien paikannukseen tarvittiin useita eri toteutusteknologioita. Tässä luvussa pohditaan, miten käyttötiedonkeruu näillä menetelmillä onnistui, mitä jatkokehityksiä UCL:ään tulisi tehdä ja miltä näyttää yleisesti lähitulevaisuus tämän työn aiheen osalta.

6.1 Päätelaite- ja paikkatiedonkeruun onnistuminen

Paikannuksen osalta UCL:ssä päädyttiin siis kolmeen vaihtoehtoon: Geolocation API, IP-paikannus, katuosoitteen perusteella tehtävä paikannus. Näistä W3C:n Geolocation API hyödyntää luvussa 4 esiteltyjä paikannusvaihtoehto tehokkaasti. Niinpä sillä saadut tulokset ovat erittäin hyviä ja varsinkin WLAN:n ansiosta päätelaitteet saadaan paikannettua tarkasti edellyttäen, että päätelaitteen selain tukee Geolocation API:a. IP-paikannus on näistä kolmesta epätarkin vaihtoehto, sillä sen paikannustarkkuus riittää vain kaupunkikohtaiseen tulokseen, jolloin eri kaupunginosista paikannus päättyy kaupungin keskustaan. Samaan lopputulokseen päästään Geolocation API:lla silloin, kun päätelaite käyttää langallista lähiverkkoa. Kaupunkikohtainen tulos on kuitenkin suuntaa-antava paikkatieto, jota voi tarvittaessa UCL:ssä siirtää kartalla haluamaansa paikkaan. Katuosoitteella saadaan niin ikään hyviä tuloksia, mutta se saaminen lomakkeelle ei ole automatisoitu. Niinpä käyttäjien täytyy syöttää se itse joka kerta UCL:ään. Tämä on työlästä matkapuhelimilla, joita ei saada muilla keinoilla paikannettua, lukuun ottamatta Iphone, jonka Safari selain tukee Geolocation API:a. Haasteeksi jäi siis edelleen paikkatiedon saaminen matkapuhelimista, joissa paikkatiedon arkaluonteisuuden lisäksi ongelmia aiheutti GPS:n puuttuminen useista päätelaitteista. Tähän voi kuitenkin olla tulossa ratkaisu hyvinkin pian, sillä GPS:n ja etenkin A-GPS:n yleistymisen on havaittavissa. Nokia toi A-GPS:n minikannettavaansa Booklet 3G:n¹ yhteydessä syksyllä 2009. Lisäksi Geolocation API:n tukeminen tulee varmasti kasvattamaan suosiotansa W3C:n, helppokäyttöisyytensä ja paikannustarkkuutensa ansiosta. Paikannuksen tulevaisuuden näkymistä on tarkemmin luvussa 6.3. UCL:n jatkokehitystä ja uusien sovellusten toteuttamista ajatellen Geolocation API:n testaaminen ja käyttöönotto toivat arvokasta tietoa.

¹<http://www.nokia.fi/tuotteet/minikannettavat/nokia-booklet-3g>

Päätelaitetunnistuksessa onnistuttiin varsinkin matkapuhelimien osalta paremmin kuin paikkatiedonkeräämisessä. Myöskään päätelaitetunnistuksessa ratkaisua laitetietojen saamiseen ei onnistuttu rajaamaan yhteen toteutusteknologiaan. Drupalin Mobile Plugin² toimii hyvin päätelaitteen selaimen tunnistamiseen, mutta varsinaisia laitetietoja sillä saadaan ainoastaan puhelinten omien selaimien avulla. Tällöin päästään hakemaan tiedot WURFL:sta³, josta saadaan käytännössä samat tiedot kuin saataisiin laitevalmistajien tarjoamien päätelaitteprofiilien eli RDF-dokumenttien avulla. Tämä on kuitenkin huomattavasti työläämpi toteutusvaihtoehto, sillä WURFL tarjoaa puhelinten tiedot keskitetysti yhdessä paikassa. Opera Miniä ja työpöytäselainta käyttävien laitteiden tietoihin WURFL ei kuitenkaan sovellu. Näitä varten täytyy ottaa käyttöön \$_SERVER-muuttuja, jolla voidaan lukea Opera Minin lähettämistä HTTP-otsikkotiedoista päätelaitteen tietoja. Työpöytäselaimille ratkaisuna on puolestaan PHP:n get_browser()-funktio, jonka käyttöönottoon Drupalissa on tehty Browscap-moduuli⁴. Näiden lisäksi päätelaitteen resoluutio täytyy lukea erikseen JavaScriptin avulla. Niinpä laitetunnistuksen optimiratkaisu on yhdistellä kartoitettuja teknologioita, jotta mahdollisimman monesta laitteesta saataisiin päätelaitetiedot.

6.2 Jatkokehitysideoita

UCL-verkkoympäristöä toteuttaessa ei kaikkia suunniteltuja ominaisuuksia ollut mahdollista toteuttaa projektin aikana, eikä osalle toiminnoista nähty tarvetta muun muassa opiskelijoiden viikkotehtävien kannalta. Lisäksi jo projektin aikana ilmestyi jatkuvasti uusia puhelimia ja mobiiliselaimia markkinoille. Uusimpien puhelimien lisäksi myös muilla kuin Nokian puhelimilla testaamista tulisi tehdä UCL:ssä, jotta laitetunnistuksen toimivuudesta voitaisiin tehdä yleisemmät johtopäätökset. Uusien mobiiliselainten toimivuus UCL:ssä jäi projektin aikana selvittämättä, sillä esimerkiksi Opera Minin versio 5 ja Opera Mobile 10 julkaistiin maaliskuussa 2010 (Opera Software, 2010a). Näitäkin potentiaalisemmalta selaimelta vaikuttaa Firefox Mobile⁵, joka tunnetaan myös nimellä Fennec⁶. Selain on tarkoitettu lähinnä Maemo-alustalle, jota edustaa muun muassa Nokia N900, mutta selain toimii myös osassa Android- ja Windows Mobile -puhelimia (Wiki, 2010). UCL:n kannalta selaimesta tekee mielenkiintoisen mahdollinen tuki Geolocation API:lle⁷, jota käytetään yhtenä paikannusvaihtoehtona UCL-verkkoympäristössä. Sekä Firefox Mobile että Geolocation API ovat Mozillan tuotteita, joten voidaan olettaa, että tuki ennen pitkää tulee

²<http://drupal.org/project/mobileplugin>

³<http://wurfl.sourceforge.net>

⁴<http://drupal.org/project/browscap>

⁵<http://www.mozilla.com/en-US/mobile>

⁶<https://wiki.mozilla.org/Fennec>

⁷<http://www.mozilla.com/fi/firefox/geolocation>

myös mobiiliselaimen. Tämä siitä syystä, että Firefox Mobile käyttää samaa selainmoottoria kuin Mozillan työpöytäselain Firefox 3.6, jossa on tuki Geolocation API:lle (Wiki, 2010). Tuen yleistyminen matkapuhelimiin ratkaisisi mobiililaitteiden paikannuksen UCL:ssä, sillä sen todettiin toimivat hyvin WLAN:ia ja Iphonea käytettäessä.

Toteutuksen jälkeen ovat muutamat Drupal-moduulit päivittyneet, kuten laite-tunnistukseen käytetty Mobile Plugin⁸, eikä kaikkia ole UCL-verkkoympäristöön päivitetty toteutuksen ja hypermedian opiskelijoiden viikkoharjoitusten jälkeen. Samoin WURFL:n⁹ laitetietokanta eli XML-dokumentti päivittyy tasaisin väliajoin yhteisön korjauksien ja uusien laitetietojen ansiosta. Aivan uusimpia puhelimia ei UCL:n käyttökokemuksien kirjaamiseen käytetty, mutta WURFL:n päivittämisellä ja Nokia RDA:n¹⁰ avulla päätelaitetunnistusta testaamalla voitaisiin varautua uusimpien laitteiden käyttäytymiseen UCL-verkkoympäristössä. Myös uusia moduuleita ilmestyy jatkuvasti, joten niissä on hyvin todennäköisesti lisäominaisuuksia verkkoympäristöön. Sellainen on esimerkiksi Nokia Mobile Theme¹¹ matkapuhelimille tarkoitettu teema, joka perustuu UCL:n käyttämään Mobile Pluginiin. Teeman teossa ovat auttaneet WURFL:n kehittäjä Andrea Trasatti ja Forum Nokia ja sen luvataan toimivan Nokioiden lisäksi myös kaikissa muissa mobiililaitteissa.

Yksi kehittämisen kohde olisi alkuperäisten päätelaitetietojen saaminen talteen, kun käyttäjä avaa käyttökokemuslomakkeen selaimellaan. Nyt tämä "arvaus" saadaan työpöytänäkymässä talteen vasta, kun käyttäjä on tallentanut lomakkeen UCL:ään. Mobiilinäkymässä tekniset tiedot ovat piilotettuina, joten päätelaitetunnistuksella saadut oletusarvot tallentuvat sellaisenaan. Tällöin käyttökokemusten versioita vertailemalla nähdään muokkauksien erot eli se, mitkä tiedot tunnistuksessa menivät väärin. Yksi vaihtoehto olisi piilottaa tekniset tiedot myös työpöytäselaimen näkymästä, kun ei muokata kertaalleen tallennettua käyttökokemusta. Tämä lisäisi käyttäjien tarvetta muokata käyttökokemuslomakkeen tietoja, mikä puolestaan saattaisi aiheuttaa paheksuntaa käyttäjissä.

Myös muutamissa UCL:n perustoiminnoissa olisi jatkokehitykselle tarvetta, sillä esimerkiksi käyttökokemuksien määrän kasvaessa myös palvelut, avainsanat ja yleisesti koko tietokanta kasvaa entisestään. Niinpä verkkoympäristöön tulisi toteuttaa hakutoiminto, jolla voisi listauksien selaamista nopeammin löytää esimerkiksi mistä palveluista käyttökokemuksia on kirjattu ja hakea käyttökokemuksia avainsanojen mukaisesti. Avainsanojen kirjaamiseen puolestaan tarvittaisiin selkeämmät ohjeet, sillä yksittäisten sanojen sijaan osa käyttäjistä kirjasi avainsanoiksi kokonaisia lauseita. Tämä luo tagipilvistä suuria ja siten vaikeasti luettavia sekä vaikeuttaa

⁸<http://drupal.org/project/mobileplugin>

⁹<http://wurfl.sourceforge.net>

¹⁰<http://www.forum.nokia.com/rda>

¹¹http://drupal.org/project/nokia_mobile

käyttökokemusten yhtenäistämistä. Avainsanoista muodostuu liian yksilöllisiä, eikä niitä voi muut käyttäjät välttämättä hyödyntää. Ohjeistusta ja ylipäättänsä koko UCL:n käyttötarkoitusta voisi selventää käyttäjille esittelyvideon avulla. Kätevimmin tämä onnistuu Youtuben¹² ja Nokia RDA:n avulla, sillä RDA:lla on mahdollista nauhoittaa matkapuhelimen käyttöä, jonka voi ladata aputeksteineen Youtubeen ja tuoda video sieltä esimerkiksi UCL:n etusivulle. Videoiden ja kuvien hyödyntäminen käyttökokemusten kirjaamisessa puolestaan rikastaisi entisestään verkkopalveluiden käytöstä kerättyjä tietoja. Niiden avulla voisi osoittaa avainsanoja ja kommentteja paremmin esimerkiksi käyttötilanteeseen vaikuttavia tekijöitä, käytön onnistumista tai epäonnistumista. Tiedostojen tallentaminen käyttökokemuksiin on nytkin mahdollista, mutta ominaisuutta ei ole ohjeistettu eikä hyödynnetty toteutuksessa.

Monipuolisempaa laitetestausta tulisi tehdä käyttökokemuslomakkeen toimimisen testaamiseksi, sillä ainakin osassa Nokian puhelimista Nokian oma selain kaatui usein ilman virheilmoitusta. Mitään yhtä selkeätä syytä ei tähän löydetty, mutta ongelma johtuu todennäköisesti liian raskaasta websivusta. Nokian selaimen latauspalkin mukaan käyttökokemuslomakkeen sivu on muutamia satoja kilotavuja, joten tämä luultavasti on liikaa muutamien Nokian mallien muistimäärälle (vrt. Lehto, 2009). Yhtenä syynä sivun raskaaseen sisältöön ovat käytetyt kuvakkeet, joita sivulla on kymmeniä. Ratkaisuksi ongelmaan voisi kokeilla Sprite-grafikan¹³ käyttöä. Nykyisessä toteutuksessa jokainen kuvake on erillinen kuvatiedosto, mutta Spritessä kaikki kuvakkeet ovat ryhmiteltyinä yhteen kuvatiedostoon. Tyylitiedostossa määritellään kuvatiedoston nimi ja se kohta, jossa tietty kuvake sijaitsee esimerkiksi pikselin tarkkuudella. Opera Minillä ei vastaavanlaisia ongelmia ilmennyt ollenkaan, mutta siirtomäärien pienentämiseksi ja siten websivun nopeuttamiseksi Sprite olisi toimiva tekniikka. (Svidgen.com, s.a.)

6.3 Tulevaisuuden näkymiä

Tutkimusyhtiö Gartner¹⁴ (2010) listaa lehdistötiedotteessansa mobiiliteknologioiden näkymiä vuosille 2010 ja 2011. Kymmenen listalla on useampi UCL:n kannalta mielenkiintoinen sovellus ja teknologia. Gartnerin mukaan yritysten kannattaa seurata mobiilin Internetin kehitystä, sillä vuoteen 2011 mennessä älypuhelimista noin 60 prosentissa on selain, jolla pystyy entistä paremmin käyttämään perinteisiä HTML-sivustoja. Tämän lisäksi älypuhelinien kehittyvät näytöt ja suurempi resoluutio nostattavat verkkosivujen käytön suosiota mobiililaitteilla. Kuten myös tämän työn luvussa 3.1 todetaan, kannattaa yritysten siis seurata, millaisilla mobiililaitteilla ja mihin tarkoitukseen heidän websivustoansa käytetään. Tällöin voidaan tarjota mo-

¹²<http://www.youtube.com>

¹³http://en.wikipedia.org/wiki/Sprite_%28computer_graphics%29 [viitattu: 20.4.2010]

¹⁴<http://www.gartner.com>

biilikäyttäjille kohdistettua sisältöä sekä parantaa verkkopalvelun saavutettavuutta ja käytettävyyttä. Myös kosketusnäyttöjen osuus kasvaa entisestään, sillä arviolta yli 60 prosentissa Euroopan ja Pohjois-Amerikan mobiililaitteista on kosketusnäyttö vuonna 2011. Tämän vuoksi kosketusnäytölliset laitteet on otettava jatkossa paremmin huomioon verkkopalveluissa, jotta niiden käyttökokemus paranee.

Seuraavana Gartnerin (2010) listalla on mobiili widgetit¹⁵, jotka tunnetaan myös nimellä vimpain (Linnake, 2007). Vimpaimen avulla matkapuhelimen oletusnäykseen eli niin sanottuun kotinäyttöön voidaan asentaa kevyt websovellus, joka näyttää verkkopalvelun sisältöä esimerkiksi JavaScriptin tai HTML:n avulla. Gartnerin mukaan vimpaimet kehittyvät nopeasti puhelinten ja käyttöjärjestelmien mukana. UCL:n kannalta vimpain olisi kätevä lisä, sillä se mahdollistaisi käyttäjille nopeamman pääsyn UCL:ään. Samalla se voisi lisätä UCL:n käyttöä, koska vimpaimen avulla UCL:n sisältö olisi paremmin esillä.

Gartnerin (2010) mukaan käyttöjärjestelmien kirjo ei ainakaan vähene lähivuosina. Niinpä markkinoilla voi olla jopa viisi kilpailevaa käyttöjärjestelmää. Tämän vuoksi sovelluskehittäjille on tärkeää, että on olemassa ohjelmistotyökaluja, jotka helpottavat sovellusten yhtäaikaista tekemistä eri käyttöjärjestelmillä. Tällainen työkalu olisi ehdottoman tärkeä, jotta esimerkiksi paikannukseen tarkoitettu sovellus toimisi mahdollisimman monessa laitteissa ilman, että jokaiselle käyttöjärjestelmälle täytyisi koodata oma sovelluksensa. Sovelluskehittäjien näkökulmasta sovellusten esille saamisen edellytykset paranevat, kun seuraavien vuosien aikana Applen Store¹⁶, Nokia Ovi¹⁷ ja Googlen Android Market¹⁸ kasvattavat merkitystään sovellusten jakamisessa asiakkaille.

Myös matkapuhelin- ja satelliittipaikannusverkoissa tapahtuu muutoksia tulevien vuosien aikana. Langaton laajakaista yleistyy ja ennen kaikkea nopeutuu entisestään, ja 3G-yhteyksistä on tulossa vakio-ominaisuus etenkin yrityskannettaviin. Suomeen ja muualle Eurooppaan aletaan rakentaa LTE-verkkoja jo vuoden 2010 lopulla (Lehto, 2010b). Lisäksi muutoksia on havaittavissa etenkin paikannuksessa, sillä kuten tässäkin työssä todetaan, paikannuspalvelut ja -sovellukset kehittyvät. Vuoden 2011 loppuun mennessä peräti 75 prosentissa matkapuhelimia on GPS-paikannin. Myös langattomiin lähiverkkoihin ja tukiasemasoluihin perustuvat paikannusmenetelmät kehittyvät ja pysyvät tärkeinä tilanteissa, joissa GPS:ää ei ole saatavilla tai se on epäluotettava. Paikannuksen kehittyessä tulee kuitenkin edelleen ottaa huomioon käyttäjien yksityisyys siten, ettei käyttäjän ole pakko suostua paikannettavaksi sovellusten toimesta. (vrt. Lehto, 2010a)

¹⁵http://en.wikipedia.org/wiki/Widget_engine#Mobile_widgets [viitattu: 23.4.2010]

¹⁶<http://store.apple.com>

¹⁷<http://www.ovi.com>

¹⁸<http://www.android.com/market>

Myös tutkimusyhtiö Berg Insightin¹⁹ tutkimuksessa paikannuksen tulevaisuutta pidetään hyvänä. Yhtiön mukaan vuonna 2009 myytiin GPS-tekniikalla varusteltuja puhelimia 150 miljoonaa kappaletta, mikä tarkoittaa peräti 92 prosentin kasvua. Vastaavanlaisen tahdin ennustetaan jatkuvan, joten yhtiön arvioissa vuonna 2014 GPS-sirullisia laitteita olisi 770 miljoonaa. Tämä tarkoittaa, että paikannuksesta tulee perustekniikka nykyisten älypuhelimien lisäksi myös perusmalleihin, mikä lisää sijaintinsa tietävien kuluttajien määrää huomattavasti. Berg Insight mukaan paikannustarkkuus paranee huomattavasti rinnakkaisten järjestelmien avulla, sillä tämä parantaa sekä tarkkuutta että toimintavarmuutta. Yhtiö uskoo, että vuoden 2011 aikana kauppoihin ilmestyy venäläisen GLONASS-paikannusjärjestelmän ja GPS:n yhdistelmää hyödyntävät älypuhelimet. Paikannustarkkuus paranee tästä entisestään, kun myös eurooppalainen Galileo-paikannusjärjestelmä saadaan käyttöön. (Pitkänen, 2010b; Pitkänen, 2010a) Kaiken kaikkiaan paikannuksen tulevaisuus näyttää siis erittäin mielenkiintoiselta, sillä paikannusjärjestelmien kehitys mahdollistaa myös tarkat tulokset sisätiloissa. Tähän kun vielä lisätään Geolocation API:n yleistyminen, niin näkymät ovat sovelluskehittäjän kannalta erittäin hyvät.

¹⁹<http://www.berginsight.com>

7. YHTEENVETO

Tässä työssä on luotu UCL:n ja kartoituksen avulla pohja päätelaite- ja paikkatiedon keräämiselle ja sen eri toteutusvaihtoehdoille. Kartoituksesta tuli laaja, sillä projektin alussa ei mahdollisista toteutusteknologioista tiedetty paljoakaan. Tutkimuksen teoriaosuudessa kartoitettiin ensin laitteistokeruun menetelmiä, joilla laitetietoja on mahdollista saada kerättyä automatisoidusti. Vaihtoehtoja löytyy useita, mutta valinta oli UCL:n kohdalla lopulta selvä, sillä WURFL tarjosi täysin ilmaiseksi suuren laitteistotietokannan. Lisäksi UCL:n toteutusalueksi valittuun Drupaliin löytyy valmis Mobile Plugin -moduuli, joka käyttää WURFL:ia laitetunnistuksessa. Tämä ei kuitenkaan riitä kuin tehokkaaseen matkapuhelinten tunnistamiseen, joten työpöytäselaimia varten täytyy ottaa käyttöön muita vaihtoehtoja. PHP:n avulla käytönotettavat `$_SERVER`-muuttuja ja `get_browser()`-funktio tarjoavat tehokkaan ratkaisun työpöytäselaintunnistukseen. Jälkimmäisen käyttöönottamista helpottaa Drupal-ympäristössä Browscap-moduuli, jonka avulla on helppo ylläpitää funktion vaatimaa browscap.ini-tiedostoa.

Jotta mahdollisimman monesta laitteesta saataisiin päätelaitetietoja automatisoidusti kerättyä, tulee siis ottaa käyttöön yhdistelmä eri menetelmiä. Edellä mainituilla ratkaisuilla on mahdollista kerätä matkapuhelimista WURFL:n tarjoamat tiedot, jotka ovat käytännössä samat kuin UAProf-sanaston avulla kuvailtavat päätelaitetiedot valmistajien RDF-dokumenteissa. Matkapuhelimista saadaan siis UCL:ään käyttökokemuksiin liitettyvien tietojen, kuten puhelimen merkin, mallin, selaimen ja näytön koon lisäksi muun muassa julkaisuaikajankohda, näppäimistön asettelu, maksimi- ja minimikoko kuville sekä useiden kuva- ja ääniformaattien tukitiedot. Opera Miniä käyttävien laitteiden tiedot saadaan puolestaan luettua `$_SERVER`-muuttujalla selaimen omista HTTP-otsikkotiedoista. Työpöytäselaimista, kuten Firefoxista, Operasta, Internet Explorerista ja Safarista saadaan selaimen nimi, versio, päätelaitteen käyttöjärjestelmä sekä JavaScriptillä päätelaitteen näytön koko. Sen sijaan ei löytynyt menetelmää, jolla olisi saatu kerättyä työpöytäselainta käyttävän päätelaitteen merkki ja malli.

Paikkatietoa ei yhdelläkään edellä mainituilla tiedonkeruuteknologialla saada, joten tämän tiedon keräämistä varten täytyy ottaa käyttöön muut keinot. Myös paikkatiedon tapauksessa ongelmaksi muodostuu päätelaitteiden runsas kirjo, joten sijaintia ei saada kaikista laitteista vain yhtä menetelmää käyttäen. Tutkimuksen toi-

nen laajempi kokonaisuus sisälsi paikannuksen toteuttamisvaihtoehtojen kartoittamista. Työpöytäselain osalta ylivoimaisesti paras ja tehokkain ratkaisu on käyttää Geolocation API:a, jonka paikannustarkkuus on WLAN:ia käytettäessä muutamista metreistä muutamaa kymmeniin metreihin. Jos päätelaite käyttää langallista lähiverkkoa, paikannustarkkuudessa päästään vain kaupunkitasolle, mikä tarkoittaa jopa kymmenien kilometrien epätarkkuutta tuloksissa. Toimintoa ei tue vielä moni selain, mutta tuki laajee varmasti W3C:n ja toiminnon käyttökelpoisuuden ansiosta. Muut työpöytäselaimet saadaan paikannettua automatisoidusti ainoastaan IP-paikannuksella, joka voidaan toteuttaa esimerkiksi hostip.info-verkkosivuston API:n avulla. Tämän lisäksi UCL:ään otettiin käyttöön myös katuosoitteen perusteella tehtävä paikannus, joka toimii ainoastaan käyttäjien syötteiden mukaisesti, eikä siten ole automatisoitu menetelmä. Puhelimista ainoastaan Iphonen Safari-selain tukee Geolocation API:a, joten puhelinten paikantamiseen täytyy etsiä muita ratkaisuja. UCL:n toteutuksessa ei löydetty yksinkertaista ja ilmaista ratkaisua puhelinten paikannukselle. Solupaikannuskirjastot ovat ainakin toistaiseksi sen verran pieniä, ettei solutiedoilla päästä järkeviin paikannustuloksiin tarpeeksi usein. Myöskään puhelimiin asennettavista sovelluksista ei ollut ratkaisuksi paikannukseen, sillä eri alustojen määrä ja sovellusten allekirjoittamisen tarve muodostuivat esteiksi.

Automatisoituun päätelaite- ja paikkatiedon keräämiseen löytyy siis useita eri menetelmiä, joista Drupaliin parhaiten soveltuvat valittiin UCL-verkkoympäristön toteutusteknologioiksi. Päätelaitetietojen automatisoitu kerääminen osoittautuu tässä työssä esiteltyjen testauksien mukaisesti mahdolliseksi. Toteutukseen tarvitaan kuitenkin useita menetelmiä, joiden valintaa tulee pohtia verkkopalvelun toteutustalustan näkökulmasta. Laitteistotiedonkeruukirjastojen tarjoamat eri API:t mahdollistavat hyvinkin erilaisia toteutuksia, joita ei tässä työssä vertailtu keskenään. Samoin laitetunnistusmenetelmistä maksamalla voi saada enemmän hyötyä dataa analysoidessa muun muassa erilaisilla visualisoinneilla ja laskelmilla. Lisäksi maksulliset lisenssit mahdollistavat kaupallisen palvelun toteuttamisen sekä tietokannan tietojen muokkaamisen omaan käyttöön. Vastaavasti paikannuksessa tarvittavat lisenssit hankkiessa voi paikkatiedon saada puhelimesta talteen erillisillä sovelluksilla. Valmiiden maksuttomien ja maksullisten sovellusten määrä on valtava, eikä tämän työn puitteissa testattu kuin muutamaa niistä. Niinpä ratkaisu automatisoidulle paikantamiselle saattaa löytyä valmiista sovelluksista. Lisäksi tämän työn tuloksia tulee vertailla uusien teknologioiden, joita kehitellään jatkuvasti. Etenkin mobiilialustojen ja paikannuksen tulevaisuuden näkymät tulevat varmasti parantamaan tässä työssä esiteltyjä tuloksia. GPS-vastaanottimien yleistymisen matkapuhelimissa, Galileo-paikannusjärjestelmän käyttöönottoaminen ja Geolocation API:n laajentaminen useampaan selaimen lisäävät paikannuspalveluiden kysyntää ja parantavat paikannustarkkuutta entisestään.

LÄHTEET

- Abbott, C., & Wilberg, N. H. (s.a.). *DetectRight - how does it compare?* Lainattu 18.1.2010, saatavilla http://www.detectright.com/docs/DetectRight_Compared.pdf
- Apostolakis, C. (2008, 21. joulukuuta). *Baeoo APIs*. Lainattu 4.3.2010, saatavilla <http://www.baeoo.com/mainslide.asp?page=developers.asp>
- Arne, J. (2008). *mobiForge Blog: Useful X headers*. Lainattu 12.12.2009, saatavilla <http://mobiforge.com/developing/blog/useful-x-headers>
- Babin, L., Good, N. A., Kromann, F. M., & Stephens, J. (2005). *PHP 5 Recipes A Problem-Solution Approach*. Apress 2560 Ninth Street, Suite 219, Berkley, CA94710: A-Press.
- Balakrishnan, M., Mohomed, I., & Ramasubramanian, V. (2009). *Where's that phone?: geolocating IP addresses on 3G networks* (Tekn. rap.). Microsoft Research Silicon Valley Mountain View, CA 94043: Microsoft Research. Lainattu 13.3.2010, saatavilla <http://doi.acm.org/10.1145/1644893.1644928>
- Bandekar, M. (2005, 16. marraskuuta). *Opera Mini Preview*. Lainattu 12.12.2009, saatavilla http://www.techtree.com/India/Reviews/Opera_Mini_Preview/551-69208-598-2.html
- DetectRight.com. (s.a.-a). *DR API*. Lainattu 12.12.2009, saatavilla <http://www.detectright.com/page/api>
- DetectRight.com. (s.a.-b). *Features*. Lainattu 12.12.2009, saatavilla <http://www.detectright.com/page/features>
- DetectRight.com. (s.a.-c). *Nokia N73 specifications from DetectRight.com*. Lainattu 12.12.2009, saatavilla <http://www.detectright.com/Device/Nokia/N73>
- Developer's Home. (s.a.). *Using UAProf (User Agent Profile) to Detect User Agent Types and Device Capabilities*. Lainattu 18.1.2010, saatavilla <http://www.developershome.com/wap/detection/detection.asp?page=uaprof>
- DeviceAtlas. (s.a.-a). *About DeviceAtlas*. Lainattu 12.12.2009, saatavilla <http://deviceatlas.com/about>
- DeviceAtlas. (s.a.-b). *DeviceAtlas Products*. Lainattu 12.12.2009, saatavilla <http://deviceatlas.com/licences>

- Digitoday. (2010, 13. huhtikuuta). *Opera-selain pääsi iPhoneen*. Lainattu 14.4.2010, saatavilla <http://www.digitoday.fi/data/2010/04/13/opera-selain-paasi-iphoneen/20105211/66>
- Djuknic, G. M., & Richton, R. E. (2001). *Geolocation and Assisted GPS* (Tekn. rap.). Atlanta, GA: Bell Laboratories, Lucent Technologies. Lainattu 15.2.2010, saatavilla <http://doi.ieeecomputersociety.org/10.1109/2.901174>
- Drupal. (2009, 13. elokuuta). *Theming and going Mobile: drupal@upei*. Lainattu 2.3.2010, saatavilla <http://drupal.org/node/534852>
- Eriksson, E. (2006). *Developer case study: Managing Java fragmentation, Opera Software's Java ME browser client*. Lainattu 12.12.2009, saatavilla http://developer.sonyericsson.com/site/global/newsandevents/latestnews/newsjune06/p_opera_mini_java_casestudy.jsp
- Etherington, D. (2009, 1. kesäkuuta). *TheAppleBlog - Apple Bringing Geolocation to Mobile Safari*. Lainattu 4.3.2010, saatavilla <http://theappleblog.com/2009/06/01/apple-bringing-geolocation-to-mobile-safari>
- Forum Nokia. (s.a.-a). *Capability granting process*. Lainattu 19.4.2010, saatavilla http://www.forum.nokia.com/Technology_Topics/Application_Quality/Testing/Capability_Granting_Process
- Forum Nokia. (s.a.-b). *Instructions for RDA usage*. Lainattu 15.3.2010, saatavilla http://www.forum.nokia.com/Technology_Topics/Application_Quality/Testing/Remote_Device_Access/Instructions_for_RDA_Usage.xhtml
- Forum Nokia. (s.a.-c). *Remote Device Access Services*. Lainattu 15.3.2010, saatavilla <http://www.forum.nokia.com/rda>
- Forum Nokia. (2009, 13. marraskuuta). *Remote Device Access for Dummies*. Lainattu 15.3.2010, saatavilla http://sw.nokia.com/id/318e7448-7a54-4d08-84c2-e4ea556a528c/RDA_for_Dummies_v1_1_en.pdf
- Forum Nokia Library. (s.a.). *Java Platform Micro Edition (Java ME)*. Lainattu 22.3.2010, saatavilla http://library.forum.nokia.com/index.jsp?topic=/Java_Developers_Library/GUID-FA71975B-D944-416E-B2E5-7B30D49BCF04.html
- Forum Nokia Wiki. (2009a, 24. lokakuuta). *Detecting Mobile Devices on Web Services*. Lainattu 12.12.2009, saatavilla http://wiki.forum.nokia.com/index.php?title=Detecting_Mobile_Devices_on_Web_Services&oldid=65784

- Forum Nokia Wiki. (2009b, 30. syyskuuta). *Java ME FAQ*. Lainattu 18.4.2010, saatavilla http://wiki.forum.nokia.com/index.php?title=Java_ME_FAQ&oldid=64934#Can_I_use_a_certificate_and_key_created_by_me_to_sign_my_midlets.3F
- Forum Nokia Wiki. (2009c, 8. kesäkuuta). *Java ME signing for dummies*. Lainattu 18.4.2010, saatavilla http://wiki.forum.nokia.com/index.php?title=Java_ME_signing_for_dummies&oldid=51555
- Forum Nokia Wiki. (2009d, 27. elokuuta). *Remote Device Access*. Lainattu 15.3.2010, saatavilla http://wiki.forum.nokia.com/index.php?title=Remote_Device_Access&diff=58331&oldid=43227
- Forum Nokia Wiki. (2009e, 5. syyskuuta). *Using built-in GPS and JavaScript to display your current position on Google Maps*. Lainattu 22.3.2010, saatavilla http://wiki.forum.nokia.com/index.php?title=Using_built-in_GPS_and_JavaScript_to_display_your_current_position_on_Google_Maps&diff=59602&oldid=59601
- Forum Nokia Wiki. (2009f, 24. syyskuuta). *Verisign signing - Java ME*. Lainattu 18.4.2010, saatavilla http://wiki.forum.nokia.com/index.php?title=Verisign_signing_-_Java_ME&oldid=63662
- Forum Nokia Wiki. (2010, 12. helmikuuta). *RDA device list*. Lainattu 16.3.2010, saatavilla http://wiki.forum.nokia.com/index.php?title=RDA_device_list&diff=71194&oldid=71193
- Google Code. (s.a.). *Geolocation API*. Lainattu 15.2.2010, saatavilla http://code.google.com/intl/fi/apis/gears/api_geolocation.html
- GPS Daily. (2010, 18. tammikuuta). *China Launches Third Orbiter For Indigenous Global SatNav System*. Lainattu 12.3.2010, saatavilla http://www.gpsdaily.com/reports/China_Launches_Third_Orbiter_For_Indigenous_Global_SatNav_System_999.html
- Handset Detection. (s.a.-a). *Full Feature List*. Lainattu 12.12.2009, saatavilla <http://www.handsetdetection.com/features/full-feature-list>
- Handset Detection. (s.a.-b). *How Handset Detection Works*. Lainattu 12.12.2009, saatavilla <http://www.handsetdetection.com/features/how-it-works>
- Handset Detection. (s.a.-c). *Nokia N73*. Lainattu 12.12.2009, saatavilla http://www.handsetdetection.com/devices/properties/Nokia/nokia_n73_ver1

- Handset Detection. (s.a.-d). *Pricing*. Lainattu 12.12.2009, saatavilla <http://www.handsetdetection.com/pricing>
- Hätäkeskuslaitos. (s.a.). *Matkapuhelinpaikannus*. Lainattu 11.2.2010, saatavilla <http://www.hatakeskus.fi/index.php?pageName=matkapuhelinpaikannus>
- Huhtamäki, J. (2005). *Rakenteisen tiedon hallittu julkaisuprosessi* [Diplomityö]. Lainattu 25.4.2010, saatavilla http://matriisi.ee.tut.fi/hypermedia/julkaisut/di_jukka_huhtamaki.pdf
- Innanen, A., & Saarimäki, J. (2009). *Internet-oikeus*. Helsinki: Edita Publishing Oy.
- Keith, G. (s.a.). *Browser Capabilities Project*. Lainattu 12.12.2009, saatavilla <http://browsers.garykeith.com>
- Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M. H., et al. (2004, 15. tammikuuta). *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0, W3C Recommendation*. Lainattu 25.4.2010, saatavilla <http://www.w3.org/TR/CCPP-struct-vocab>
- Kotilainen, S. (2010, 1. tammikuuta). *Tietokone - Pahasti myöhässä: Galileo-paikannus vihdoin rakenteille*. Lainattu 8.3.2010, saatavilla http://www.tietokone.fi/uutiset/pahasti_myohassa_galileo_paikannus_vihdoin_rakenteille
- Kummala, J., Kulmala, R., Hautala, R., Oinas, J., & Holm, C. (2001). *Matkapuhelinpaikannuksen hyödyntäminen liikennetietojen keruussa* (Esiselvitys - Tietohallinnon selvityksiä 61/2001). Oy Edita Ab, Helsinki: Tiehallinto. Lainattu 5.2.2010, saatavilla <http://alk.tiehallinto.fi/julkaisut/pdf/3200707.pdf>
- Lardinois, F. (2010, 4. maaliskuuta). *Google Chrome Becomes Location Aware*. Lainattu 13.4.2010, saatavilla http://www.readwriteweb.com/archives/google_chrome_becomes_location_aware.php
- Lehto, T. (2009, 6. tammikuuta). *Nokia, suurin tietokonevalmistaja*. Lainattu 20.4.2010, saatavilla <http://lehto.net/blogi/2009/01/nokia-suurin-tietokonevalmistaja.htm>
- Lehto, T. (2010a, 24. maaliskuuta). *Gartner Outlines 10 Mobile Technologies to Watch in 2010 and 2011*. Lainattu 23.4.2010, saatavilla <http://www.gartner.com/it/page.jsp?id=1328113>

- Lehto, T. (2010b, 13. huhtikuuta). *Sonera avaa 4g-verkkonsa jo tänä vuonna*. Lainattu 23.4.2010, saatavilla http://www.tietokone.fi/uutiset/sonera_avaa_4g_verkkonsa_jo_tana_vuonna
- Lähteinen, L., & Tervakari, A.-M. (2010). *Käyttäjien suhtautuminen UCL-ympäristöön*.
- Linnake, T. (2007, 14. syyskuuta). *Telkku.comin ja Ylen vimpaimet suosiossa kännyköissä*. Lainattu 20.4.2010, saatavilla <http://www.itviikko.fi/uutiset/2007/09/14/telkkucomin-ja-ylen-vimpaimet-suosiossa-kannykoissa/200722553/7>
- Mahmoud, Q. H. (2004, maaliskuuta). *J2ME and Location-Based Services*. Lainattu 18.4.2010, saatavilla <http://developers.sun.com/mobility/apis/articles/location>
- Malik, O. (2010, 10. tammikuuta). *GigaOM - Will 2010 Finally Be the Year of Location?* Lainattu 8.3.2010, saatavilla <http://gigaom.com/2010/01/10/2010-year-of-location>
- Mapsi. (s.a.). *Mikä on Mapsi?* Lainattu 7.3.2010, saatavilla <http://www.mapsi.fi/mika-on-mapsi>
- Mapsi. (2009, 17. heinäkuuta). *Yleiskarttana OpenStreetMap*. Lainattu 7.3.2010, saatavilla <http://www.mapsi.fi/mapsin-yleiskarttana-on-openstreetmap>
- McLachlan, J. (s.a.-a). *mDevInf – Mobile Device Information*. Lainattu 12.12.2009, saatavilla http://wurfl.sourceforge.net/utilities/m_dev_inf.php
- McLachlan, J. (s.a.-b). *Mobile Device Information*. Lainattu 12.12.2009, saatavilla <http://mdevinf.sourceforge.net>
- McQuillan, M. (2009). *mobiForge Blog: Testing Mobile Web Sites Using Firefox*. Lainattu 17.3.2010, saatavilla <http://mobiforge.com/testing/story/testing-mobile-web-sites-using-firefox>
- Miettinen, S. (2006). *GPS käsikirja*. Helsinki: Genimap Oy.
- Mikkonen, T. (2004). *Mobiiliohjelmointi*. Helsinki: Talentum.
- Mozilla. (s.a.). *Selaa sijaintisi mukaan*. Lainattu 15.2.2010, saatavilla <http://www.mozilla.com/fi/firefox/geolocation>
- Mozilla. (2010). *Mozilla Firefox Privacy Policy*. Lainattu 15.2.2010, saatavilla <http://www.mozilla.com/en-US/legal/privacy/firefox-en.html>

- Mozilla Developer Center. (2010). *Using geolocation*. Lainattu 15.2.2010, saatavilla https://developer.mozilla.org/En/Using_geolocation
- NCSA HTTPd. (2008). *CGI Environment Variables*. Lainattu 12.12.2009, saatavilla <http://hoohoo.ncsa.illinois.edu/cgi/env.html>
- Negrino, T., & Smith, D. (2007). *JavaScript - Tehokasta hallintaa*. Helsinki: Read-me.fi.
- Nelson, G. (2010, 12. tammikuuta). *CIO - Is 2010 the year of location-based services?* Lainattu 8.3.2010, saatavilla <http://www.cio.co.uk/article/3210035/is-2010-the-year-of-location-based-services>
- Nokia. (s.a.-a). *Assisted GPS*. Lainattu 28.2.2010, saatavilla <http://www.nokia-asia.com/explore-services/nokia-maps-new/support/nokia-maps-2-0/assisted-gps>
- Nokia. (s.a.-b). *User Guide - Nokia Sports Tracker - Beta Version 0.9*. Lainattu 6.3.2010, saatavilla http://sportstracker.nokia.com/nts/Nokia_Sports_Tracker_User_Guide.pdf
- Nokia. (2008, 3. joulukuuta). *Java ME Quick Start*. Lainattu 23.3.2010, saatavilla http://heanet.dl.sourceforge.net/project/pys60/pys60/1.4.5/PythonForS60_1_4_5_doc.pdf
- Open Source at Nokia. (2010, 17. helmikuuta). *Python for S60*. Lainattu 23.3.2010, saatavilla http://wiki.opensource.nokia.com/index.php?title=Python_for_S60&diff=3678&oldid=3581
- Openwave Developer Network. (s.a.). *Introducing WURFL*. Lainattu 12.12.2009, saatavilla http://developer.openwave.com/dvl/tools_and_sdk/wurfl_and_wall/intro_wurfl.htm
- Opera Software. (2009). *Opera Software: Third quarter 2009*. Lainattu 12.12.2009, saatavilla http://www.opera.com/media/finance/2009/3Q09_presentation.pdf
- Opera Software. (2010a, 16. maaliskuuta). *No mobile phone left behind: Opera Mini 5 and Opera Mobile 10 introduced in final, consumer-ready versions*. Lainattu 19.4.2010, saatavilla <http://www.opera.com/press/releases/2010/03/16>
- Opera Software. (2010b, 12. huhtikuuta). *Opera browsers exceed 100 million users*. Lainattu 12.4.2010, saatavilla <http://www.opera.com/press/releases/2010/04/12>

- Paikannus.com. (s.a.-a). *Langattomiin lähiverkkoihin perustuva paikannus*. Lainattu 9.3.2010, saatavilla <http://www.paikannus.com/langattomiin-lahiverkkoihin-perustuva-paikannus>
- Paikannus.com. (s.a.-b). *Matkapuhelinverkkoon perustuva paikannus*. Lainattu 3.2.2010, saatavilla <http://www.paikannus.com/matkapuhelinverkkoon-perustuva-paikannus>
- Paikannus.com. (s.a.-c). *Satelliittipaikannus*. Lainattu 28.2.2010, saatavilla <http://www.paikannus.com/matkapuhelinverkkoon-perustuva-paikannus>
- Palmu, V. (2008, 15. toukokuuta). *Suomalaiset Drupal-saitit*. Lainattu 9.4.2010, saatavilla <http://drupal.fi/fi/drupal-suomessa/suomalaiset-drupal-saitit>
- Passani, L. (s.a.-a). *Welcome to the new WURFL DB*. Lainattu 12.12.2009, saatavilla http://www.wurflpro.com/static/new_wurflpdb.htm
- Passani, L. (s.a.-b). *What about UAProf?* Lainattu 18.1.2010, saatavilla <http://wurfl.sourceforge.net/uaprof.php>
- Passani, L. (s.a.-c). *WURFL*. Lainattu 12.12.2009, saatavilla <http://wurfl.sourceforge.net>
- Peltoniemi, J. (2010, 10. maaliskuuta). *PROSESSORI - Käteen sopiva WLAN-mittari*. Lainattu 12.3.2010, saatavilla <http://www.proessori.fi/uutiset/uutinen2.asp?id=55365>
- Penttinen, J. (2006). *Tietoliikennetekniikka 3G ja erityisverkot*. Helsinki: WSOY.
- Pitkänen, J. (2010a, 23. huhtikuuta). *Berg Insight predicts 894 million mobile banking users by 2015*. Lainattu 23.4.2010, saatavilla http://www.berginsight.com/News.aspx?m_m=6&s_m=1
- Pitkänen, J. (2010b, 22. maaliskuuta). *Paikannuksen suosio kasvaa puhelimissa*. Lainattu 23.4.2010, saatavilla http://www.tietokone.fi/uutiset/paikannuksen-suosio_kasvaa_puhelimissa
- Pulli, H., Posti, A., & Tapaninen, U. (2009). *Tukke - Tuoteseuranta satamasi-donnaisessa kuljetusketjussa* (Turun yliopiston merenkulkualan koulutus- ja tutkimuskeskuksen julkaisuja). Painosalama Oy, Turku: Turun yliopisto. Lainattu 5.2.2010, saatavilla <https://oa.doria.fi/bitstream/handle/10024/47155/MKKB167.pdf?sequence=1>

- Rainio, A. (2003). *Paikannus mobiilipalveluissa ja sovelluksissa* (Teknologiakatsaus 143/2003). Helsinki: TEKES. Lainattu 3.2.2010, saatavilla http://www.tekes.fi/fi/document/43291/paikannus_mobiilipalveluissa_pdf
- Ronda, T., & Bila, N. (s.a.). *Appono: A Geolocator for the Internet* (Tekn. rap.). Toronto, ON: University of Toronto. Lainattu 12.3.2010, saatavilla http://www.cs.toronto.edu/~ronda/courses/internet_systems/project/appono_report.pdf
- Salonen, J., & Huhtamäki, J. (2010). *Launching Context-Aware Visualisations*. Proceedings of OPAALS 2010, 22-23 March 2010, Aracaju, SE, Brazil.
- Sani, I. (2008, 25. tammikuuta). *Moduulit: valinnaisia toimintoja ja lisäosia*. Lainattu 25.4.2010, saatavilla <http://drupal.fi/fi/kayttajan-opas/moduulit-valinnaisia-toimintoja-ja-lisaosia>
- Sanoma News. (2008, 11. heinäkuuta). *Digitodayn uutiset iPhoneen*. Lainattu 8.4.2010, saatavilla <http://www.sanomaneews.com/Content.aspx?d=10111>
- Silius, K., Tervakari, A.-M., & Miilumäki, T. (2009). *Learning by doing as mobile with a Urban Computing Lab*. Proceedings of International Technology Enhanced Learning Conference 2009 (TELearn 2009) on the 6-8 October 2009 Taipei, Taiwan. CD-ROM. Lainattu 3.12.2010, saatavilla <http://matriisi.ee.tut.fi/hypermedia//julkaisut/2009-silius-et-al-learningbydoing-ucl.pdf>
- Sipilä, J. (2010, 4. maaliskuuta). *MTV3 - Näin satelliittipaikannus selvitti Bobcat-varkauden*. Lainattu 8.3.2010, saatavilla <http://www.mtv3.fi/uutiset/rikos.shtml/arkistot/rikos/2010/03/1070755>
- Storey, D. (2007, 31. elokuuta). *Evolving the Internet on your phone: Designing web sites with Opera Mini 4 in mind*. Lainattu 25.4.2010, saatavilla <http://dev.opera.com/articles/view/evolving-the-internet-on-your-phone-des>
- Streng Hähre, K. von. (2007, 7. marraskuuta). *Opera Mini request headers*. Lainattu 12.12.2009, saatavilla <http://dev.opera.com/articles/view/opera-mini-request-headers>
- Svidgen.com. (s.a.). *CSS Sprites Demo*. Lainattu 20.4.2010, saatavilla <http://www.svidgen.com/sprites>
- Symbian Foundation. (2010a, 10. maaliskuuta). *Java ME Quick Start*. Lainattu 22.3.2010, saatavilla http://developer.symbian.org/wiki/index.php?title=Java_ME_Quick_Start&diff=46607&oldid=24823

- Symbian Foundation. (2010b, 10. maaliskuuta). *Python in a Nutshell*. Lainattu 22.3.2010, saatavilla http://developer.symbian.org/wiki/index.php?title=Python_in_a_Nutshell&diff=46637&oldid=25646
- Tampereen yliopisto. (2010, 4. maaliskuuta). *Tampereen yliopisto mukana Mapsi-mobiilikarttapalvelussa*. Lainattu 7.3.2010, saatavilla <http://www.uta.fi/opiskelu/uutiset/ilmoitukset.php?ilmoitus=46035>
- The PHP Group. (s.a.). *PHP Manual – get_browser*. Lainattu 12.12.2009, saatavilla <http://php.net/manual/en/function.get-browser.php>
- Tietokone. (2010, 14. huhtikuuta). *Testaa kannyselaimesi, vaihda vikkelämpään*. Lainattu 14.4.2010, saatavilla http://www.tietokone.fi/uutiset/testaa_kannyselaimesi_vaihda_vikkelampaan
- Viestintävirasto. (s.a.). *Sanastoa*. Lainattu 12.3.2010, saatavilla <http://www.ficora.fi/index/palvelut/palvelutaiheittain/puhelinjalaajakaista/uudistukset.html>
- Virtanen, A., & Koskinen, S. (2005). *NOPPA Näkövammaisten opetusjärjestelmän pilottiprojekti* (Tekn. rap.). Tampere: VTT Tuotteet ja tuotanto. Lainattu 19.3.2010, saatavilla http://www.elsa.fi/tietopankki/tietopankki_pdf/Noppa%20loppuraportti%20final.pdf
- W3C. (s.a.). *Geolocation Working Group Overview*. Lainattu 4.3.2010, saatavilla <http://www.w3.org/2008/geolocation>
- W3Schools. (s.a.-a). *ASP Server Variables Collection*. Lainattu 12.12.2009, saatavilla http://www.w3schools.com/asp/coll_servervariables.asp
- W3Schools. (s.a.-b). *PHP get_browser() Function*. Lainattu 12.12.2009, saatavilla http://www.w3schools.com/PHP/func_misc_get_browser.asp
- Wiki, M. (2010, 13. huhtikuuta). *Mobile Platforms and Projects*. Lainattu 25.4.2010, saatavilla <https://wiki.mozilla.org/index.php?title=Mobile/Platforms&oldid=215161>
- Wikipedia. (2010, 3. huhtikuuta). *Windows XP Professional x64 Edition*. Lainattu 25.4.2010, saatavilla http://en.wikipedia.org/w/index.php?title=Windows_XP_Professional_x64_Edition&oldid=353743495
- Wireless Application Protocol Forum. (2001, 20. lokakuuta). *WAG UAProf - Version 20-Oct-2001*. Lainattu 18.1.2010, saatavilla <http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf>

LIITE 1: MOBILE PLUGIN -ESIMERKKEJÄ

Esimerkki 1: Mobile Pluginin laitetunnistus, Nokia N73

```
HTTP_USER_AGENT: NokiaN73-1/2.0628.0.0.1 S60/3.0
Profile/MIDP-2.0 Configuration/CLDC-1.1
ID: nokia_n73_ver1_20628001
brand_name: Nokia
model_name: N73
marketing_name:
model_extra_info:
unique: true
uniqueness_handler:
is_wireless_device: true
device_claims_web_support: true
has_qwerty_keyboard: false
can_skip_aligned_link_row: true
uaprof: http://nds.nokia.com/uaprof/NN73-1r100.xml
uaprof2: http://nds1.nds.nokia.com/uaprof/NN73r100.xml
uaprof3:
nokia_series: 60
nokia_edition: 3
pointing_method: joystick
mobile_browser: Nokia
mobile_browser_version:
preferred_markup: html_wi_oma_xhtmlmp_1_0
resolution_width: 240
resolution_height: 320
```

Esimerkki 2: Mobile Pluginin selaintunnistus, Firefox 3.5.5

```
HTTP_USER_AGENT: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR 3.5.30729)
ID: firefox_20b2_beosbecp_enus
brand_name: generic web browser
model_name: 2.0
marketing_name:
model_extra_info:
unique: true
uniqueness_handler:
is_wireless_device: false
device_claims_web_support: true
has_qwerty_keyboard: true
```

```
can_skip_aligned_link_row: true
uaprof:
uaprof2:
uaprof3:
nokia_series: 0
nokia_edition: 0
pointing_method: touchscreen
mobile_browser:
mobile_browser_version:
preferred_markup: html_web_4_0
resolution_width: 770
resolution_height: 300
```

Esimerkki 3: Mobile Pluginin selaintunnistus, Opera Mini 4.2

```
HTTP_USER_AGENT:
Opera/9.80 (J2ME/MIDP; Opera Mini/4.2.14912/1186; U; en)
Presto/2.2.0
ID: opera_mini_ver4
brand_name: Opera
model_name: Mini
marketing_name:
model_extra_info:
unique: false
uniqueness_handler:
is_wireless_device: true
device_claims_web_support: false
has_qwerty_keyboard: false
can_skip_aligned_link_row: false
uaprof:
uaprof2:
uaprof3:
nokia_series: 0
nokia_edition: 0
pointing_method:
mobile_browser: Opera Mini
mobile_browser_version: 4
preferred_markup: html_wi_oma_xhtmlmp_1_0
resolution_width: 128
resolution_height: 92
```

LIITE 2: BROWSCAP-ESIMERKKEJÄ

Esimerkki 1: Browscap-moduulin laitetunnistus, Nokia N73

```
Array
(
  [browser] => Nokia [version] => 0 [majorver] => 0
  [minorver] => 0 [platform] => unknown [alpha] => [beta] =>
  [win16] => [win32] => [win64] => [frames] => [iframes] =>
  [tables] => 1 [cookies] => 1 [backgroundsounds] =>
  [cdf] => [vbscript] => [javaapplets] => [javascript] =>
  [activexcontrols] => [isbanned] => [ismobiledevice] => 1
  [issyndicationreader] => [crawler] => [cssversion] => 0
  [supportscss] => [aol] => [aolversion] => 0 [parent] =>
  Nokia [useragent] => NokiaN73-1/2.0628.0.0.1 S60/3.0
  Profile/MIDP-2.0 Configuration/CLDC-1.1
  [browser_name_pattern] => *Nokia*/*
)
```

Esimerkki 2: Browscap-moduulin selaintunnistus, Firefox 3.5.5

```
Array
(
  [browser] => Firefox [version] => 3.5 [majorver] => 3
  [minorver] => 5 [platform] => WinXP [alpha] =>
  [beta] => [win16] => [win32] => 1 [win64] =>
  [frames] => 1 [iframes] => 1 [tables] => 1
  [cookies] => 1 [backgroundsounds] => [cdf] =>
  [vbscript] => [javaapplets] => 1 [javascript] => 1
  [activexcontrols] => [isbanned] => [ismobiledevice] =>
  [issyndicationreader] => [crawler] => [cssversion] => 3
  [supportscss] => 1 [aol] => [aolversion] => 0
  [parent] => Firefox 3.5 [useragent] => Mozilla/5.0
  (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.5) Gecko/20091102
  Firefox/3.5.5 (.NET CLR 3.5.30729)
  [browser_name_pattern] => Mozilla/5.0 (Windows; *;
  Windows NT 5.1; *; rv:1.9.*) Gecko/* Firefox/3.5*
)
```

Esimerkki 3: Browscap-moduulin selaintunnistus, Opera Mini 4.2

```
Array
(
[browser] => Opera Mini [version] => 4.2 [majorver] => 4
[minorver] => 2 [platform] => unknown [alpha] =>
[beta] => [win16] => [win32] => [win64] => [frames] => 1
[iframes] => 1 [tables] => 1 [cookies] => 1
[backgroundsounds] => [cdf] => [vbscript] => [javaapplets] =>
[javascript] => 1 [activexcontrols] => [isbanned] =>
[ismobiledevice] => 1 [issyndicationreader] => [crawler] =>
[cssversion] => 0[supportscss] => [aol] => [aolversion] => 0
[parent] => Opera Mini [useragent] => Opera/9.80
(J2ME/MIDP; Opera Mini/4.2.14912/1186; U; en) Presto/2.2.0
[browser_name_pattern] => Opera/* (J2ME/MIDP; Opera Mini/4.2*)*)
)
```

LIITE 3: PÄÄTELAITETUNNISTUKSEN TULOKSIA UCL:SSÄ

PUHELINTUNNISTUKSET:

Nokia E65

Päätelaite: Matkapuhelin

Puhelinmerkki: Nokia

Puhelinmalli: E65

Selain: -

Selaimen versio: -

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 240

Resoluutio (korkeus): 320

Nokia E65 (Opera Mini 4.2)

Päätelaite: Matkapuhelin

Puhelinmerkki: Nokia

Puhelinmalli: E65

Selain: Opera Mini

Selaimen versio: 4

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 128

Resoluutio (korkeus): 92

Nokia N97

Päätelaite: Matkapuhelin

Puhelinmerkki: Nokia

Selain: Nokia

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 176

Resoluutio (korkeus): 208

Nokia E75

Päätelaite: Matkapuhelin

Puhelinmerkki: Nokia

Puhelinmalli: E75

Selain: Nokia

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 176

Resoluutio (korkeus): 208

Nokia E70

Päätelaitte: Matkapuhelin

Puhelinmerkki: Nokia

Puhelinmalli: E70

Selain: Safari

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 352

Resoluutio (korkeus): 416

Nokia E70 (Opera Mini 4.2)

Päätelaitte: Matkapuhelin

Puhelinmerkki: Nokia

Puhelinmalli: E70

Selain: Opera Mini

Selaimen versio: 4

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 128

Resoluutio (korkeus): 92

Nokia 5500 Sport (Opera Mini 4.2)

Päätelaitte: Matkapuhelin

Puhelinmerkki: Nokia

Puhelinmalli: 5500

Selain: Opera Mini

Selaimen versio: 4

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 128

Resoluutio (korkeus): 92

Nokia 5500 Sport

Päätelaitte: Matkapuhelin

Puhelinmerkki: Nokia

Puhelinmalli: 5500

Selain: Nokia

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 208

Resoluutio (korkeus): 208

Nokia 6700 slide

Päätelaitte: Matkapuhelin

Puhelinmerkki: Nokia

Selain: Nokia

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 176

Resoluutio (korkeus): 208

Nokia 6700 slide (Opera Mini 4.2)

Päätelaitte: Matkapuhelin

Puhelinmerkki: Nokia

Selain: Opera Mini

Selaimen versio: 4

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 128

Resoluutio (korkeus): 92

Nokia N96

Päätelaitte: Matkapuhelin

Puhelinmerkki: Nokia

Puhelinmalli: N96

Selain: Safari

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 240

Resoluutio (korkeus): 320

Nokia N96 (Opera Mini 4.2)

Päätelaitte: Matkapuhelin

Puhelinmerkki: Nokia

Selain: Opera Mini

Selaimen versio: 4

Yhteystyyppi: Matkapuhelinverkko (2G/3G)

Resoluutio (leveys): 128

Resoluutio (korkeus): 92

Opera Mini Demo (4.2) Päätelaitte: Matkapuhelin

Puhelinmerkki: Opera

Puhelinmalli: Mini
Selain: Opera Mini
Selaimen versio: 4
Yhteystyyppi: Matkapuhelinverkko (2G/3G)
Resoluutio (leveys): 128
Resoluutio (korkeus): 92

TYÖPÖYTÄKONETUNNISTUKSET:

Firefox 3.5.2 WinXP

Päätelaite: -
Käyttöjärjestelmä: WinXP
Selain: Firefox
Selaimen versio: 3.5
Yhteystyyppi: -
Resoluutio (leveys): 1280
Resoluutio (korkeus): 1024

Firefox 3.6.3 WinXP 64bit

Päätelaite: -
Käyttöjärjestelmä: Win2003
Selain: Firefox
Selaimen versio: 3.6
Yhteystyyppi: -
Resoluutio (leveys): 1680
Resoluutio (korkeus): 1050

Internet Explorer 8 WinXP 64bit

Päätelaite: -
Käyttöjärjestelmä: Win2003
Selain: IE
Selaimen versio: 8
Yhteystyyppi: -
Resoluutio (leveys): -
Resoluutio (korkeus): -

Chrome 4.0 WinXP

Päätelaite: -
Käyttöjärjestelmä: WinXP

Selain: Chrome

Selaimen versio: 4.0

Yhteystyyppi: -

Resoluutio (leveys): -

Resoluutio (korkeus): -

Opera 10 Win7

Päätelaitte: -

Käyttöjärjestelmä: Win7

Selain: Opera

Selaimen versio: 10

Yhteystyyppi: -

Resoluutio (leveys): -

Resoluutio (korkeus): -

Firefox 3.0.10 Ubuntu 8.04 (hardy)

Päätelaitte: -

Käyttöjärjestelmä: Linux

Selain: Firefox

Selaimen versio: 3.0

Yhteystyyppi: -

Resoluutio (leveys): -

Resoluutio (korkeus): -