
Hirvivaara-widgetin kääntäminen Androidilta Symbian³-alustalle

Lasse Koponen

Opinnäytetyö

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Lasse Koponen	
Työn nimi Hirvivaara-widgetin kääntäminen Androidilta Symbian^3-alustalle	
Päiväys 26.4.2011	Sivumäärä/Liitteet 28
Ohjaaja(t) Jussi Koistinen, tietotekniikan lehtori, Sami Lahti, tietotekniikan lehtori	
Toimeksiantaja/Yhteistyökumppani(t) Mobie Oy, toimitusjohtaja Kari Vatanen, ohjelmistosuunnittelija Joni Eskelinen	
Tiivistelmä <p>Tämän opinnäytetyön aiheena oli Android-käyttöjärjestelmälle suunnitellun widget-ohjelman kääntäminen Symbian^3-järjestelmälle. Opinnäytetyön tilasi Mobie Oy.</p> <p>Opinnäytetyössä kuvataan käännösprosessi, joka sisälsi käännettävään sovelluksen sekä Android-käyttöjärjestelmän evaluoinnin, perehtymisen Symbian^3-järjestelmään sekä sille käytettävissä olevien sovelluskehitystekniikoiden arvioinnin ja lopullisen toteutustekniikan valinnan. Työssä tutustuttiin tarkemmin Qt- sekä Web Runtime -tekniikoihin, joiden avulla itse käännössovellus toteutettiin. Työn aikana tutkittiin myös Android- ja Symbian^3-käyttöjärjestelmien eroja sovelluskehityksen näkökulmasta.</p> <p>Työ sisälsi sovelluksen suunnittelu-, toteutus- sekä yksikkötestausvaiheet. Tuloksena opinnäytetyöstä saatiin vaatimusten mukaisesti toimiva sovellus. Sovelluksen lopullinen testaus sekä julkaisu suoritetaan työn jälkeisenä ajankohtana.</p>	
Avainsanat Android, Symbian, widget	
Luottamuksellisuus julkinen	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Computer Science			
Author(s) Lasse Koponen			
Title of Thesis Converting Android-widget to Symbian^3-Operating System			
Date	26 April 2011	Pages/Appendices	28
Supervisor(s) Jussi Koistinen, Sami Lahti			
Project/Partners Mobie Oy, Kari Vatanen, Joni Eskelinen			
<p>Abstract</p> <p>The aim of this final project was to convert an Android-widget to the Symbian^3-operating system. The project was commissioned by Mobie Oy.</p> <p>The first phase of the project was to get to know the operating systems in question. Next, it had to be decided, which techniques would be used to make the conversion. In this case, Qt-framework and Web Runtime-framework were chosen to be used in making the conversion. Next step was getting familiar with these frameworks and designing the application. After that started the actual coding of the application. Basic testing routines were executed during the coding process. The final testing of the application will commence after the timeframe of this final project.</p> <p>In this final project the differences between Android and Symbian^3 operating systems were evaluated from the application developer's perspective, too.</p>			
Keywords Android, Symbian, widget			
Confidentiality public			

Alkusanat

Tämä opinnäytetyö on tehty Savonia-ammattikorkeakoulun tietotekniikan koulutusohjelman opinnäytetyönä keväällä 2011. Työn tilasi Mobie Oy. Työn ohjasivat tietotekniikan lehtorit Jussi Koistinen ja Sami Lahti Savonia-ammattikorkeakoulusta. Mobie Oy:stä opinnäytetyötä ohjasivat toimitusjohtaja Kari Vatanen sekä ohjelmistosuunnittelija Joni Eskelinen. Haluan kiittää kaikkia, jotka osallistuivat työn ohjaukseen ja mahdollistivat sen valmistumisen.

Kuopiossa 26.4.2011

Lasse Koponen

SISÄLTÖ

TERMIT JA LYHENTEET	6
1 JOHDANTO	7
2 SYMBIAN^3-KÄYTTÖJÄRJESTELMÄ	9
2.1 Qt	11
2.2 Web Runtime.....	13
3 ANDROID-KÄYTTÖJÄRJESTELMÄ	16
4 SYMBIAN^3:N JA ANDROIDIN EROT WIDGET-PIENOISOHJELMAN TOTEUTUKSESSA....	18
4.1 Widget-piensohjelman idea.....	18
4.2 Tutustuminen Symbian-alustan mahdollisuuksiin.....	19
4.3 Työkalut.....	19
4.3.1 Eclipse SDK, Aptana Studio Plug-In sekä Nokia WRT Plug-In.....	20
4.3.2 Qt Creator.....	21
5 WIDGETIN TOTEUTUS SYMBIAN^3-ALUSTALLE.....	24
6 TESTAAMINEN.....	25
7 YHTEENVETO	26
LÄHTEET	28

TERMIT JA LYHENTEET

Symbian^3	Nokian uusimmissa kosketusnäyttöpuhelimissa käytettävä käyttöjärjestelmä.
Android	Googlen kehittämä käyttöjärjestelmä, joka on suunnattu nykyaikaisille älypuhelimille.
Qt	Ohjelmistojen ja graafisten käyttöliittymien kehitysympäristö, jota tukevat mm. Symbian-käyttöjärjestelmät.
WRT	Web Runtime. Nokian kehittämä tekniikka, jolla älypuhelimiin voidaan kehittää web-sovelluksia.
Widget	PC-tietokoneissa sekä nykyaikaisissa älypuhelimissa käytetty pienen ohjelma.
C++	C-ohjelmointikielestä kehitetty kieli, jossa on olio-ohjelmoinnin mahdollisuus.
SQLite	Relaatiotietokantajärjestelmä, jossa tietokanta voidaan tallentaa yhteen tiedostoon. Soveltuu erityisesti sovellusohjelmien kanssa käytettäväksi.
HTML	Hyper Text Markup Language. Kuvauskieli, jolla on toteutettu nykyaikaiset Internet-sivustot.
JavaScript	Komentosarjakieli, jota käytetään nykyaikaisissa web-sivustoissa dynaamisen toiminnallisuuden toteuttamiseen.
HTTP	Hyper Text Transfer Protocol
GPS	Global Positioning System. Satelliitteihin perustuva paikannusjärjestelmä.
XML	eXtensible Markup Language. Merkintäkieli, jonka avulla voidaan merkitä tiedon tarkoitus itse tiedon yhteyteen.

1 JOHDANTO

Mobiilisovellukset ovat yleistyneet viime vuosina runsaasti älypuhelinien teknisen kehityksen sekä tästä johtuvan hintojen laskun vuoksi. Käyttäjämäärän kasvaessa myös kysyntä erilaisille mobiilisovelluksille on kasvanut.

Tämä opinnäytetyö on tehty Mobie Oy:n toimeksiannosta. Työn tavoitteena on kääntää alun perin Android-käyttöjärjestelmälle tehty widget-pienoisohjelma Symbian^3-käyttöjärjestelmällä toimivaksi.

Mobie Oy on kehittänyt Liikenneturvalle hirvivaarasta varoittavan widget-pienoissovelluksen. Tilastojen mukaan hirvionnettomuuksia tapahtuu eniten tunti aurin-
gonlaskun jälkeen. Sovellus varoittaa käyttäjää hirvivaarasta riskiaikana. Tämän lisäksi sovellus näyttää Liikenneturvan uutisotsikoita.

Mobien kehittämä sovellus on tehty Android-käyttöjärjestelmää käyttäville älypuhelimille. Kysyntä saada sovellus Nokian älypuheliiniin on ollut suurta, koska Nokian älypuhelimia on käytössä erittäin paljon. Tätä kysyntää vastaamaan on tämän opinnäytetyön puitteis-
sa tehty käännös Android-sovelluksesta Nokian älypuheliiniin. Sovelluksen luonteen vuoksi kohdealustaksi valikoitui Symbian^3-käyttöjärjestelmä, joka on ensimmäinen No-
kian älypuhelimissa käytettävä käyttöjärjestelmä, joka tukee käyttöjärjestelmän kotinäy-
töllä näytettäviä widget-pienoissovelluksia.

Opinnäytetyön alkuosassa käsitellään käännösprosessissa käytettyjä tekniikoita ja niihin liittyvää teoriaa. Tässä osassa esitellään Symbian^3-käyttöjärjestelmää yleisesti sekä käydään läpi tekniikat, joilla Symbian^3:lle voidaan kehittää sovelluksia. Tarkemmin käsi-
tellään tämän projektin sovellusalueeseen oleellisesti liittyvät WRT- sekä Qt-
ohjelmointitekniikat. Näistä tekniikoista keskitytään tarkemmin Qt:hen. Tämän jälkeen tutustutaan Android-käyttöjärjestelmään, ohjelmointitekniikoihin, joilla järjestelmälle voi-
daan kehittää sovelluksia sekä sovellusten julkaisemiseen ja jakamiseen loppukäyttäjille. Teoriaosuuden lopuksi käsitellään tämän projektin sovellusalueeseen eli widget-
ohjelmiin liittyviä eroavaisuuksia näiden kahden käyttöjärjestelmän välillä. Tässä osuu-
dessa esitellään myös sovelluskehittimet, joilla ohjelmointityö toteutettiin.

Seuraavassa osassa käsitellään projektin suunnittelu- sekä toteutusvaihetta. Ensin käsi-
tellään sovelluksen yleistä suunnittelua. Tässä vaiheessa tehdyt ratkaisut johtivat siihen, että kehitettävä sovellus hajautettiin kahdeksi erilliseksi sovellukseksi. Toteutusvaiheen

kuvauksen loppuosio jakautuukin kahteen osaan, joissa kuvataan näiden kahden sovelluksen suunnittelua ja toteutusta. Ensin tutustutaan nk. taustapalvelusovelluksen suunnitteluun sekä toteutukseen. Taustapalvelun kuvauksen jälkeen esitellään varsinaisen widget-ohjelman toteutus joka tehtiin Web Runtime(WRT) -tekniikkaa hyväksi käyttäen.

Seuraavaksi käsitellään sovelluksen testaamiseen liittyviä seikkoja, minkä jälkeen seuraa pohdintaosuus. Tässä osuudessa käsitellään projektia yleisesti ottaen, siinä käytettyjä tekniikoita ja niiden käytössä ilmenneitä niin hyviä kuin huonojakin ominaisuuksia. Lisäksi vertaillaan Android- ja Symbian³-käyttöjärjestelmiä sovelluskehittäjän näkökulmasta. Tämän jälkeen tehdään yhteenveto koko opinnäytetyöprojektistä.

2 SYMBIAN^3-KÄYTTÖJÄRJESTELMÄ

Symbian on Nokian ylläpitämä älypuhelinien käyttöjärjestelmä. Symbianin tärkeimpiä ominaisuuksia on tuki sovellusohjelmien yhtäaikaiselle suorittamiselle eli nk. moniajole. (*Symbian OS*, Forum Nokia Wiki.)

Symbian käyttöjärjestelmä on kirjoitettu C++-kielellä, mutta C++-kielen standardikirjastot on korvattu omilla, varta vasten Symbianille kirjoitetuilla kirjastoilla. Näitä kirjastoja käyttämällä pyritään pienentämään päätelaitteen resurssien kulutusta, minkä takia osa kirjastoista on toteutettu matalamman tason ohjelmoinnilla kuin C++:n standardikirjastot. (*Symbian OS*, Forum Nokia Wiki.)

Symbian^3-käyttöjärjestelmä on Nokian vuonna 2010 lanseeraama Symbian-käyttöjärjestelmän uusin versio, joka on kehitetty uusimpien älypuhelinien ominaisuuksia silmällä pitäen. Sen käyttöliittymä on optimoitu kosketusnäytöillä päätelaitteille. (*Symbian OS*, Forum Nokia Wiki)

Tiedon tallennusmuotona Symbian^3 tukee SQLite-tietokantaa. SQLite on relaatiotietokantarakenne, jossa koko tietokanta voidaan tallentaa yhteen tiedostoon. Tästä syystä SQLite on erityisen käyttökelpoinen pienempien sovellusten tiedon tallennustapana. (*About SQLite*, SQLite.)

Symbian^3-käyttöjärjestelmässä päätelaitteen ns. kotinäyttö on jaettu kolmeen käyttäjän muokattavissa olevaan työpöytä näkymään. Näille työpöydille on mahdollista lisätä widget-pienoisohjelmia sekä sovellusten pikakuvakkeita. Symbian^3 sisältää myös tuen horisontaaliselle kotinäytön esitykselle, kun päätelaite käännetään sivuttain. Alla olevassa kuvassa (KUVA 1) on esitetty Symbian^3-kotinäyttö. (*Symbian OS*, Forum Nokia Wiki.)



KUVA 1. Symbian^3-käyttöjärjestelmän kotinäyttö.

Symbian-järjestelmälle voidaan kehittää sovelluksia monilla erilaisilla tekniikoilla. Symbian^3-käyttöjärjestelmä sisältää tuen Qt-kehitysympäristölle, jota nykyään suositellaan käytettäväksi kolmansien osapuolien Symbian-sovelluksia kehitettäessä Symbian C++-kielen sijaan. Symbian^3-järjestelmä tukee myös Javan Micro Edition -alustaa, jonka avulla pystytään kehittämään Java-sovelluksia. Näiden lisäksi Nokia on kehittänyt Web Runtime -tekniikan, jonka avulla Symbian-järjestelmiin voidaan tehdä yksinkertaisia web-sovelluksia muistuttavia ohjelmia. (*Symbian*, Wikipedia.)

Sovellusten jakelu Symbian^3-käyttöjärjestelmässä on toteutettu Nokian Ovi Store -sovelluskaupan avulla. Ovi Storesta voidaan ladata niin ilmaisia kuin maksullisiakin sovelluksia. Ennen sovellusten julkaisemista Nokian Ovi Store -sovelluskaupassa sovelluk-

set, muut kuin WRT-sovellukset, tulee sertifioida Symbian Signed -ohjelman allekirjoituksella.

Sovelluksen sertifiointi tapahtuu Symbian Signed -sertifiointipalvelussa, jossa sertifioitava sovellus käy läpi tarkkojen määritysten mukaiset testit. Läpäistyään testit sovellukseen liitetään digitaalinen allekirjoitus, joka toimii varmenteena sovelluksen laadusta. Sovelluksen allekirjoitusvaatimukset riippuvat niistä päätelaitteen ominaisuuksista, joita sovellus käyttää hyväkseen. Mikäli sovellus käyttää tärkeitä päätelaitteen järjestelmäresursseja, allekirjoittamatonta sovellusta ei voi asentaa päätelaitteeseen. Mikäli sovellus käyttää vähemmän tärkeitä tietoja (esim. GPS-paikannus), sovelluksen voi asentaa allekirjoittamatta, mutta tällöin asennusohjelma varoittaa sovelluksen mahdollisesta haitallisuudesta päätelaitteeseen. Allekirjoittamalla tällaisen sovelluksen varoitusta ei esitetä asennusvaiheessa.

2.1 Qt

Qt on alun perin Trolltechin kehittämä kehitysympäristö, joka sopii ohjelmistojen ja käyttöliittymien tekemiseen monille eri ohjelmointialustoille. Qt-ympäristöllä on mahdollisuus kehittää sovelluksia useille eri kohdejärjestelmille, kuten PC-tietokoneille sekä mobiililaitteille. Lähtökohdana Qt:n kehityksessä on ollut, että samaa lähdekoodia voitaisiin käyttää mahdollisimman monessa kohdealustassa niin, että mahdollisimman suuri osa sovelluksen lähdekoodista säilyisi muuttumattomana. (Qt, Nokia.)

Qt on korkeamman tason ohjelmointikieli kuin Symbianin omat C++-kirjastot. Tästä syystä sovellusten kehittäminen Symbianille Qt:lla on yksinkertaisempaa ja huomattavasti nopeampaa kuin Symbian C++:lla. Toisaalta Qt ei tarjoa sisäänrakennettuna niin laajaa mahdollisuutta päätelaitteen ominaisuuksien hyödyntämiseen kuin matalamman tason Symbian C++ -ympäristö. Tätä varten on kehitetty Qt Mobility -kirjastot, jotka sisältävät päätelaitteiden ominaisuuksia hyödyntävät rajapinnat. Näihin rajapintoihin kuuluvat mm. paikannusjärjestelmät (GPS tai verkkopohjainen), päätelaitteen yhteystiedot sekä kalenteri. Qt Mobility -kirjaston tuorein versio on 1.1.1, joka tarjoaa rajapinnat edellä mainittujen lisäksi myös päätelaitteen värinäpalautteen antamiseen sekä kameran käsittelyyn. Tätä työtä tehtäessä käytettiin Qt Mobility -kirjaston versiota 1.0.2, koska tekohetkellä versiota 1.1.1 ei saanut käyttää kaupallisten sovellusten tekemiseen.

Nykyään Qt:n kehityksestä vastaa Trolltechistä muodostettu Qt Development Frameworks, jonka omistaa Nokia. (Qt, Nokia.)

Qt on olio-ohjelmoinnin periaatteille pohjautuva ympäristö C++-kielen tapaan. Qt:n luokkahierarkia perustuu QObject-luokkaan, jonka merkittävimpana piirteenä on mahdollisuus viestien välittämiseen eri luokkien ilmentymien välillä käyttämällä signaaleja ja slot-metodeja. QObject-luokan ilmentymän lähettämä signaali yhdistetään toisen QObject-luokan ilmentymän slot-tyyppiseen metodiin, jota kutsutaan, kun ensimmäinen QObject-ilmentymä lähettää signaalin.

QObjectin lähettämä signaali voidaan yhdistää myös saman luokan slot-metodiin. Signaalit sekä slot-metodit esitellään olio-ohjelmoinnin perusteiden mukaan luokan esittelyssä käyttämällä signals- sekä slots-erottimia, joilla ne erotetaan luokan muista jäsenfunktioista sekä jäsenmuuttujista. Kuvassa (KUVA 2) on nähtävissä esimerkkejä signaalien ja slot-metodien esittelyä.

```
signals:
    void nextRotateFeed(FeedItem item);
    void sendPreferences(QString prefs);
public slots:
    void startFeedUpdate();
    void setPreferences(QString preferences);
    void getPreferences();
```

KUVA 2. Signaalien ja slot-metodien esittelyä.

Signaalit yhdistetään slot-metodeihin käyttämällä QObject luokan staattista connect()-metodia. Alla olevassa esimerkissä (KUVA 3) server-muuttujan newConnection()-signaali yhdistetään client-muuttujan newConnection()-nimiseen slot-metodiin, jota kutsutaan aina kun server-muuttuja lähettää signaalin.

```
connect(server, SIGNAL(newConnection()), client, SLOT(newConnection()));
```

KUVA 3. Esimerkki connect-metodin käytöstä.

Periyttämällä QObject-luokkaa omiin luokkatoteutuksiin ja käyttämällä signaaleja ja slot-metodeja on mahdollista luoda luokkien välille haluttuja interaktioita ja kommunikaatiota. Qt:n käyttöliittymätoteutuksen käyttäjätapahtumat on toteutettu käyttämällä signaaleja ja slot-metodeja.

Muistinhallinta on Qt-ympäristössä toteutettu QObject-luokan ympärille. QObject-luokan perivät luokat muodostavat hierarkkisen puun. Tämä muodostetaan antamalla tietyille luokalle isäntä välittämällä isäntäluokan ilmentymä parametrina "lapsiluokan" muodostin-funktiolle. Tämän jälkeen lapsiluokan ilmentymä lisätään isäntäluokan ilmentymän lapsi-ilmentymälistaan. Tuhottaessa isäntäluokan ilmentymä tuhoaa se automaattisesti myös kaikki lapsi-ilmentymänsä, jolloin niitä ei tarvitse tuhota erikseen. Automatisointi helpottaa ohjelmoijan työtä etenkin silloin, kun luokalla on suuri määrä lapsiluokan ilmentymiä, jolloin niiden tuhoaminen käsin olisi aikaa vievää ja virhealtista. Tarvittaessa luokkahierarkiaa QObject-luokan perivien luokkien välille ei ole pakko muodostaa. Tässä tapauksessa lapsi-ilmentymien tuhoaminen täytyy tehdä käsin luokan tuhoamisfunktiossa.

Qt mahdollistaa myös natiivin ohjelmakoodin suorittamisen osana Qt-koodia. Näin voi helposti yhdistää korkeamman tason Qt-koodia matalamman tason toiminnallisuuksiin, jotka toteutetaan natiivilla koodilla. Symbian-käyttöjärjestelmän tapauksessa Qt-koodin joukkoon voi upottaa Symbian C++-kielen lauseita. Ennen suoritettavaa natiivikoodia on syytä tarkistaa kääntäjän ehtolauseilla, että ajettava ympäristö on sellainen, jossa natiivikoodia voi suorittaa.

Qt-ympäristön suurin vahvuus on sen soveltuvuus monille erilaisille alustoille työpöytäietokoneista aina nykyaikaisiin älypuhelimiin. Samaa Qt-koodia voidaan ajaa eri ympäristöissä ilman, että ohjelmakoodia tarvitsee suuresti muuttaa. Mikäli koodissa on käyttöjärjestelmäsidonaisia toiminnallisuuksia, tarvitsee vain tarkistaa, onko ajettava ympäristö oikea, ja toimia sen mukaan. (Qt. Nokia.)

2.2 Web Runtime

Web Runtime on Nokian kehittämä teknologia, jonka avulla älypuhelimille voidaan tehdä web-pohjaisia sovelluksia, jotka kuitenkin toimivat omana sovelluksenaan puhelimesta eivätkä käytä puhelimen web-selainta.

WRT-sovellukset pohjautuvat HTML-kieleen, jolla kuvataan sovelluksen rakenne. Itse sovelluslogiikka toteutetaan JavaScript-kielellä ja sovelluksen ulkoasu muokataan web-sivuillakin käytetyllä Cascading Style Sheet (CSS) -tyylitiedostolla. WRT-sovellukset poikkeavat web-sovelluksista siinä, että ne pääsevät rajapintojen kautta keskustelemaan päätelaitteissa olevien palveluiden kanssa. Näin sovelluksiin voidaan lisätä interaktiivisia

ominaisuuksia, kuten esimerkiksi GPS-pohjaista paikkatietoa sekä päätelaitteen yhteystietoja.

WRT-sovellus koostuu minimissään kahdesta tiedostosta. HTML-päätteinen tiedosto kuvaa sovelluksen rakenteen web-sivun tapaan. Tähän tiedostoon kuvataan kaikki sovelluksen näkymät sekä niiden sisältämät komponentit.

Plist-päätteinen tiedosto on XML-muotoon kirjoitettu tiedosto, jossa sovellukseen liittyvät tiedostot ja oletukset on kuvattu avain-arvo-pareilla dict-lohkon alle. Ensin tulee key-elementti, joka kuvaa sovelluksen ominaisuuden, minkä jälkeen string-elementissä kerrotaan tälle ominaisuudelle arvo. Tähän tiedostoon tulee myös tieto siitä, halutaanko sovellukseen työpöydälle sijoitettava pienoishälyttö. Tämä ilmoitetaan key-elementtiin sijoitettavalla "MiniViewEnabled" -arvolla, jonka arvoksi annetaan joko xml-elementti "true" tai "false". Seuraavassa kuvassa (KUVA 4) on nähtävissä esimerkki plist-tiedostosta. (*Create your first WRT widget*. Forum Nokia.)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Nokia//DTD PLIST 1.0//EN" "http://www.nokia.com/DTDs/plist-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>DisplayName</key>
    <string>Project</string>
    <key>Identifier</key>
    <string>com.Project.basic.widget</string>
    <key>Version</key>
    <string>1.0</string>
    <key>MainHTML</key>
    <string>index.html</string>
    <key>AllowNetworkAccess</key>
    <true/>
    <key>MiniViewEnabled</key>
    <true/>
  </dict>
</plist>
```

KUVA 4. Esimerkki plist-tiedostosta.

Näiden kahden pakollisen tiedoston lisäksi sovellukseen on mahdollista lisätä toiminnallisuutta JavaScript-tiedostoja käyttämällä. Sovelluksen ulkoasua on mahdollista muokata HTML-maailmasta tutuilla CSS-tiedostoilla. (*Create your first WRT widget*. Forum Nokia.)

Symbian^3-käyttöjärjestelmässä WRT-sovelluksiin on lisätty mahdollisuus toteuttaa sovellukseen pienoishälyttö, joka voidaan asettaa päätelaitteen työpöydälle. Pienoishälyttö toimii sovelluksen graafisena esityksenä työpöydällä eikä sisällä mahdollisuutta interaktiivisiin komponentteihin. (*Create your first WRT widget*. Forum Nokia.)

Nokia WRTKit on Nokian räätälöimä ulkoasukirjasto WRT-sovelluksia varten. Kirjasto on toteutettu JavaScriptillä, ja se koostuu erilaisista näkymä- ja käyttöliittymäkomponenteista. WRTKit-komponentit on toteutettu olio-ohjelmoinnin periaatteita hyödyntäen. Näin komponenttien käsittely ohjelmallisesti on helppoa. WRTKit tarjoaa komponenteille myös valmiin ulkoasutoteutuksen, mutta tarvittaessa komponenttien ulkoasun voi mukauttaa omien vaatimusten mukaiseksi. Kuvassa (KUVA 5) on nähtävissä WRTKit-kirjastolla luotu esimerkkikäyttöliittymä. (*WRTkit Overview, Forum Nokia.*)



KUVA 5. WRT-sovelluksen esimerkkikäyttöliittymä. Luotu käyttäen WRTKit-kirjastoa.

3 ANDROID-KÄYTTÖJÄRJESTELMÄ

Android on avoimeen lähdekoodiin perustuva, Linux-ydintä käyttävä käyttöjärjestelmä, joka on suunnattu erityisesti mobiililaitteille. Androidin kehityksestä vastasi alun perin Android Inc., mutta Googlen ostettua kyseisen yrityksen Androidin kehityksestä vastaa Open Handset Alliance. (*Android Overview*, Open Handset Alliance.)

Android-käyttöjärjestelmästä on saatavana useita versioita, ja versiot vaihtelevat päätelaitteiden välillä, siten että uudemmissa päätelaitteissa on uudempia versioita (kirjoitushetkellä versio 2.3) ja vanhemmissa laitteissa on vanhempia versioita. Päätelaitteiden valmistajat ovat vastuussa vanhojen päätelaitteiden päivittämisestä uudempaan Android-versioon.

Android-järjestelmän avoimuuden vuoksi päätelaitevalmistajat voivat muokata käyttöjärjestelmää omien tarpeidensa mukaan. Näin ovat tehneet esimerkiksi Samsung sekä HTC, joiden älypuhelimissa käytetään valmistajan omaa käyttöliittymää Androidin vakiona tarjoaman sijaan.

Android-käyttöjärjestelmän perusominaisuuksiin kuuluu käyttäjän muokattavissa olevat työpöytänäkymät. Näille työpöydille voidaan sijoittaa niin web-kirjanmerkkejä, sovellusten pikakuvakkeita kuin itsenäisiä pienoissovelluksia, nk. widgettejä.

Android-ohjelmia kehitetään pääasiassa Java-kielellä, jota ajetaan päätelaitteessa Dalvik-virtuaalikoneen päällä. Päätelaitteen ominaisuuksia on mahdollista käyttää hyväksi Googlen kehittämällä Java-kirjastoilla. Sovellusten on myös mahdollista ohittaa Dalvik-virtuaalikone, jolloin sovellusten aikakriittisiä osia, kuten graafiikan piirtäminen, voidaan toteuttaa matalamman tason ohjelmointikielellä C++:lla. Tämä lisää ohjelman suorituksen tehokkuutta.

Androidin avoin ympäristö helpottaa sovellusten toteuttamista alustalle. Ohjelmoituaan sovelluksen ohjelmoija voi lähettää sovelluksen Androidin sovelluskauppaan ladattavaksi joko ilmaiseksi tai maksua vastaan. Sovellusten laadun takeena on mahdollisuus arvostella Marketissa olevia sovelluksia. Näin laadukkaat sovellukset nostavat suosiotaan.

Avoin ympäristö tuo myös haasteita sovelluskehitykseen. Koska markkinoilla on suuri valikoima Android-käyttöjärjestelmiä ja sen eri versioita käytäviä päätelaitteita, on sovelluskehittäjän otettava huomioon mahdolliset yhteensopivuusseikat. Sovellus, joka käyttää uusimpien Android-versioiden ominaisuuksia hyväkseen, ei välttämättä toimi halutuil-

la tavalla vanhemmissa versioissa. Tämä vaikuttaa sovelluksen jakelu- ja myyntisegmentin laajuuteen. Android-järjestelmän suurin heikkous verrattaessa kilpaileviin järjestelmiin onkin järjestelmän sirpaloituminen moneen eri versioon sekä näitä vaihtelevasti tukeviin päätelaitteisiin.

Android-käyttöjärjestelmälle on saatavana runsaasti kolmansien osapuolten tekemiä sovellusohjelmia. Näiden jakaminen loppukäyttäjille on toteutettu Android Market -sovelluskaupan avulla. Suurimmassa osassa Android-käyttöjärjestelmää käyttäviä päätelaitteita on esiasennettuna "Market"-niminen sovellus, jonka avulla käyttäjät pääsevät lataamaan sovelluksia laitteeseensa. Marketissa on tarjolla niin ilmaisia kuin maksullisia-kin sovelluksia. Google sallii myös kolmansien osapuolten sovelluskauppatoimetukset, toisin kuin valtaosa kilpailevista mobiiliyrityksistä. (*Android*, Android Suomi Wiki)

4 SYMBIAN^3:N JA ANDROIDIN EROT WIDGET-PIENOISOHJELMAN TOTEUTUKSESSA

4.1 Widget-pienoisohjelman idea

Widget on pienoissovellus, jota käytetään nykyään sekä kotitietokone- että mobiilikäyttöjärjestelmissä. Widget voi olla itsenäinen sovellus tai se voi olla myös taustalla ajettavan, isomman sovelluksen inforuutu, josta käy ilmi sovelluksen tila. Esimerkkejä widget-sovelluksista ovat uutissyötesovellukset sekä musiikkisoitinsovelluksen pienoisversio, joka sisältää musiikin toiston kontrollointipainikkeet. Uusista mobiilikäyttöjärjestelmistä Android sekä Symbian^3 sisältävät mahdollisuuden asettaa pienoissovelluksia päätelaitteen työpöytänäkymään.

Tässä projektissa toteutettava Hirvivaara-widgetohjelma koostui kahdesta erillisestä toiminnallisuudesta: Widget hakee verkosta uutissyötteitä, joita kierrätetään widgetin näytössä. Tämän lisäksi widget varoittaa käyttäjää hirvivaarasta auringonlaskun jälkeen. Hirvivaaravaroitus käyttää hyväkseen päätelaitteiden paikannusjärjestelmiä. Alla olevassa kuvassa (KUVA 6) on kuvakaappaus Hirvivaara-widgetin Android-versiosta.



KUVA 6. Hirvivaara-widgetin Android-versio.

4.2 Tutustuminen Symbian-alustan mahdollisuuksiin

Widget-ohjelmien toteutus Symbian³-käyttöjärjestelmässä poikkeaa hyvin paljon Android-käyttöjärjestelmässä käytetystä toteutuksesta. Symbian-järjestelmässä ainoa teknologia, jolla widgetit voi toteuttaa, on WRT-tekniikka. Tällä tekniikalla toteutettava widget on koko näytössä toimivan sovelluksen pienennetty graafinen esitysmuoto, eikä voi sisältää itsessään mitään interaktiivisia osia. Kun työpöydällä olevaa widgetiä painetaan, avautuu sovellus koko näyttöön.

Androidissa widgetit on toteutettu periaatteessa samalla tavalla kuin Symbian³:lla jossa ne ovat sovelluksen pienennetty esitysmuoto. Android-ympäristössä widgetit ovat kuitenkin itsenäisiä ohjelmia, jotka voivat sisältää interaktiivisia osioita tavallisten koko näytössä toimivien sovellusten tapaan. Android-widgetit voivat myös käyttää kaikkia päätelaitteen ominaisuuksia rajapintojen kautta, kun taas WRT-ympäristössä sovelluksen käytössä olevat rajapinnat ovat rajallisemmat. Android-widgetin tavanomaista sovellusta muistuttava rakenne mahdollistaa myös erilaisten ajastettujen toimintojen ja prosessien käsittelyn itse widgetissä.

Sovelluksen arkkitehtuuria suunniteltaessa reunaehtona sovellukselle ollut työpöydälle sijoitettava widget-sovellus rajasi toteutustavan WRT-sovellukseen, koska se on ainoa Symbian-käyttöjärjestelmän tukemista tekniikoista, jolla työpöytäwidgetin tekeminen on mahdollista. Sovelluksen rakenne edellytti sovellukselta näkyvien ominaisuuksien lisäksi taustalla tapahtuvaa, ajastettua toiminnallisuutta. Tämän toiminnallisuuden toteuttaminen WRT-tekniikalla osoittautui mahdottomaksi, koska WRT-sovelluksen logiikka perustuu JavaScript-kieleen, jonka avulla toteutetut ajastimet pysähtyvät välittömästi, kun päätelaitteen näyttö sulkeutuu. Näin ollen ainoaksi mahdollisuudeksi jäi tehdä toinen, taustalla samanaikaisesti ajettava sovellus, joka huolehtisi ajastettujen toimintojen suorittamisesta.

Taustapalvelun toteutus päätettiin tehdä Qt-kehitysympäristöllä, koska sen tarjoamat rajapinnat Symbian³-käyttöjärjestelmän laitteisiin riittivät taustapalvelun vaatimiin toimintoihin. Qt:n käyttöä puolsi myös kehityksen helppous Symbian C++:aan nähden.

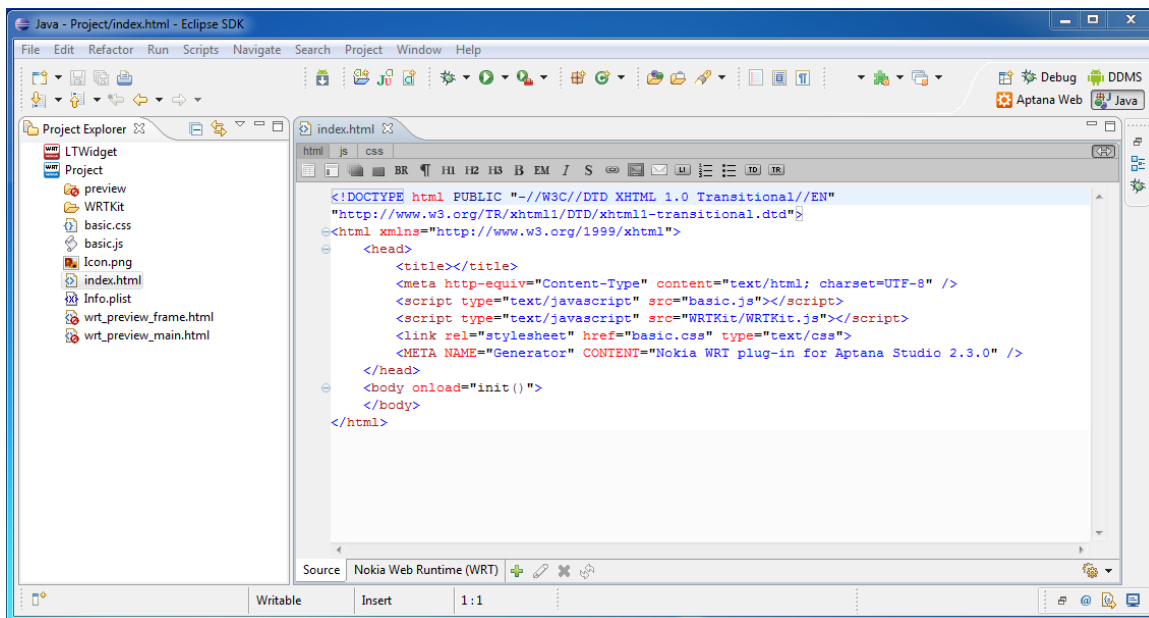
4.3 Työkalut

Toteutusta aloitettaessa tavoitteena oli ohjelmoida koko sovellus yhdellä kehitystyökalulla. Tähän tarkoitukseen soveltuvin olisi ollut Eclipse SDK, joka on ainoa laajempi sovel-

luskehitysymäristö, jolla voi toteuttaa WRT-widgettejä. Widget-ohjelmien tekemistä varten Eclipseen tulee ladata Aptana Studio Plug-In sekä tähän lisäpakettiin saatavana oleva Nokia WRT plug-in. Qt-sovellusten toteuttaminen osoittautui kuitenkin Eclipsellä niin vaivalloiseksi, että sovellus päätettiin tehdä kahdella erillisellä kehitysympäristöllä. WRT-sovellus tehtiin edelleen Eclipseen avulla, mutta taustapalveluna toimiva Qt-sovellus kehitettiin käyttämällä Nokian Qt SDK-kehityspaketin mukana tulevaa Qt Creator-ohjelmaa.

4.3.1 Eclipse SDK, Aptana Studio Plug-In sekä Nokia WRT Plug-In

Aptana Studio Plug-In on Eclipse SDK-sovelluskehitysympäristöön saatava lisäosa, joka lisää Eclipseen Aptana Studion ominaisuuksia. Näistä ominaisuuksista tässä projektissa käytettiin Aptana Studioon saatavaa Nokia WRT Plug-In-lisäosaa. WRT Plug-In:llä luotaessa uusi projekti, sovellus luo automaattisesti WRT-projektin tarvitsemat plist- sekä html-päätteiset tiedostot. Tämän lisäksi sovellus luo automaattisesti JavaScript- sekä css-tiedoston, joita voi tarvittaessa käyttää projektin kehityksessä. Kuvassa (KUVA 7) on esitetty Eclipse-sovelluskehittimen käyttöliittymä.



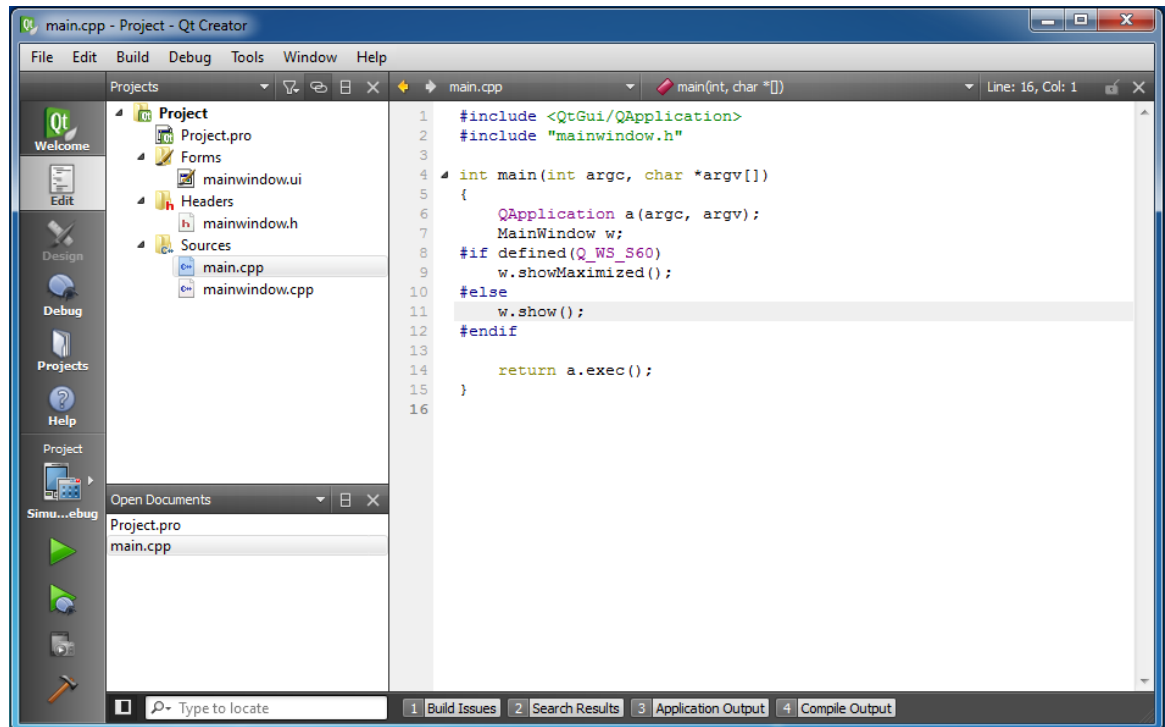
KUVA 7. WRT-projektin perusnäkyminen Eclipse SDK:ssa.

Projektin testaaminen Nokia WRT Plug-In:illa on varsin helppoa. Projektin HTML-sivun näkymässä on välilehti "Nokia Web Runtime (WRT)", joka aukaisee emulaattorisivun, jossa WRT-sovellusta voi testata. Emulaattori tarjoaa myös Symbian-käyttöjärjestelmän

kotinäytön emuloinnin, jonka avulla voi nähdä miltä sovelluksen pienoisnäyttö näyttää päätelaitteessa. Tämän helpottaa sovelluksen testaamista, koska WRT-sovellusten ovat varsin yksinkertaisia, jolloin erot emulaattoritestauksen sekä varsinaisen päätelaitteessa suoritettavan testauksen välillä ovat erittäin pieniä. Tämän vuoksi varsinkin projektin alkuvaiheessa vältetään sovelluksen siirtämiseltä päätelaitteeseen testattavaksi. Sovelluskehityksen loppuvaiheessa on tietenkin huomioitava myös varsinainen päätelaitetestaus, jotta mahdollisimman suuri osa ohjelmassa olevista virheistä voidaan havaita.

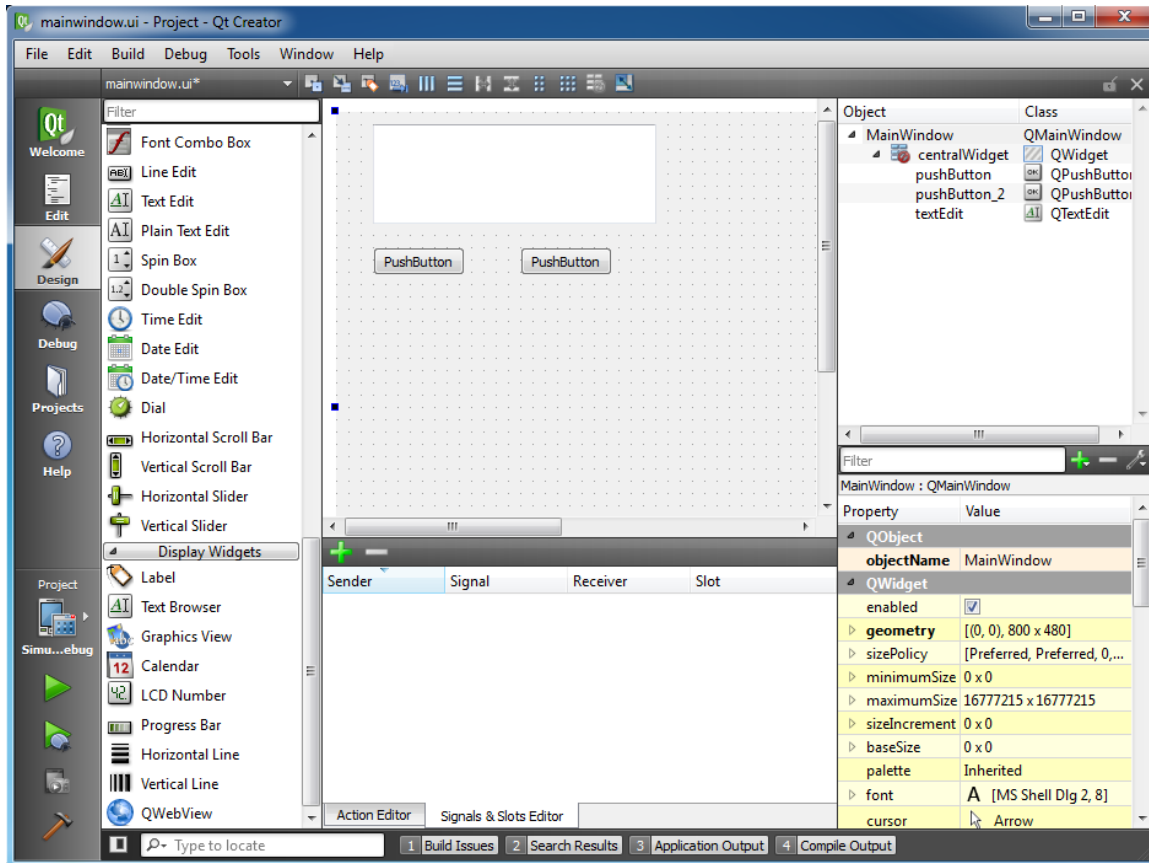
4.3.2 Qt Creator

Qt Creator-ohjelmalla kehitettäessä projektin hallinta on verrattain helppoa vaihtoehtona olleeseen Carbide C++-ohjelmaan, joka on Eclipsen-kehitysympäristön päälle rakennettu, Symbian C++-kehitykseen räätälöity ympäristö. Uutta projektia luotaessa Creator luo projektin hakemistorakenteen, josta löytyvät projektin lähdekooditiedostot sekä otsikkotiedostot eroteltuina omiin kansioihinsa. Tämän lisäksi projektin alta löytyy käännösvaiheessa tarvittava pro-päätteinen tiedosto, johon projektin sisältö on tallennettu. Creator-projektin päänäkymä on esitetty alla olevassa kuvassa (KUVA 8).



KUVA 8. Qt Creator-projektin päänäkymä.

Qt Creator sisältää käyttöliittymätyökalun, jolla sovellusten käyttöliittymien luominen on helppoa. Työkalun toiminta perustuu muista sovelluskehittimistä tuttuun tapaan, jossa käyttöliittymäkomponentteja vedetään näkyviin, minkä jälkeen komponentteihin lisätään haluttuja toiminnallisuuksia. Alla olevassa kuvassa (KUVA 9) on nähtävissä käyttöliittymätyökalun ulkoasu.



KUVA 9. Qt Creator:n käyttöliittymäeditori

Itse sovelluskoodin ohjelmointi on Qt Creator-ohjelmalla varsin sujuvaa. Ohjelma tarjoaa nykyohjelmoinnissa arkipäivää olevan ennustusmekanismin, jolla koodin kirjoittaminen on nopeaa. Ennustava koodinkirjoitus myös vähentää kirjoitusvirheitä koodissa, mikä nopeuttaa koodin saamista käännettävään muotoon. Koodieditorissa on myös aktiivinen virheentarkastus, joka kirjoitushetkellä alleviivaa virheellisesti kirjoitetun koodin. Näin virheiden korjaaminen on helppoa jo kirjoitusvaiheessa eikä vaadi erillistä koodin käännettämistä virheiden huomaamiseksi.

Käännettäessä koodia valitaan ensin kohdejärjestelmä jolle käänös tehdään. Tässä projektissa käänösmahdollisuuksina olivat käänös simulaattorihjelmalle tai Symbian-

käyttöjärjestelmälle. Käännettäessä ja ajettaessa simulaattoriversiota, Creator käynnistää simulaattorin jossa käännetty ohjelma suoritetaan. Käännettäessä Symbian-versiota Creator muodostaa sovelluskoodista sis-päätteisen asennuspaketin joka on mahdollista siirtää päätelaitteeseen asennettavaksi ja suoritettavaksi. Mikäli päätelaite on kiinnitetty tietokoneeseen, jossa ohjelmointi tehdään, käännösvaiheen jälkeen Creator siirtää ja asentaa automaattisesti sovelluksen päätelaitteeseen. Tämä jälkeen ohjelma suoritetaan päätelaitteessa.

5 WIDGETIN TOTEUTUS SYMBIAN³-ALUSTALLE

Kappaletta 5 ei voida julkaista sen sisältämien salaisten tietojen vuoksi.

6 TESTAAMINEN

Kahden erillisen sovelluksen testaaminen suoritettiin kehityksen alkuvaiheessa itsenäisesti käyttämällä kovakoodattuja syötetietoja, joilla toimintojen oikea toiminta varmistettiin. Kun kehitystyö oli edennyt riittävän pitkälle, yhdistettiin Eclipse SDK:ssa kehitetty WRT-sovellus Qt Creatorin taustapalvelun kanssa, jolloin Eclipsen WRT-sovellus lähetti komentoja Qt Creatorissa ajettulle taustapalvelulle. Tällä tavalla varmistettiin sovellusten kommunikaation toimivuus. Näin edettiin siihen asti, kunnes päästiin testaamaan oikealla päätelaitteella, jona toimi Nokia C7-00-älypuhelin.

Hirvivaaravaroituspalvelun testaaminen päätelaitteessa tehtiin sekä autolla ajaen että kävelemällä. Kävellessä hälytykseen vaadittavaa nopeuden raja-arvoa laskettiin, jotta hälytys laukeaisi myös kävelyvauhdissa, kun taas autolla ajaessa raja-arvo voitiin pitää alkuperäisessä arvossaan.

Sovelluksen päätelaitetestaus oli erittäin aikaa vievää, koska kahden erillisen sovelluksen yhteistoiminta mahdollisti sovelluksen virheellisen toiminnan hyvin erilaisissa tilanteissa. Tämän vuoksi erilaisten testitapausten luomisessa piti tuntea päätelaitteen toimintaa erilaisissa tilanteissa. Muun muassa verkkoyhteystyyppin muuttumisen testaamisessa jouduttiin etsimään testitilanne, jossa on mahdollista kokeilla siirtymät langattomasta lähiverkosta 2G- ja 3G-verkkoihin, jotta saatiin selville näiden olosuhteiden vaikutus sovelluksen toimintaan.

Opinnäytetyöprojektin puitteissa suoritettu testaus käsitti ainoastaan yksikkötestauksen. Sovelluksen laajempi kenttätestaaminen suoritetaan opinnäytetyön valmistumisen jälkeen.

7 YHTEENVETO

Tämän opinnäytetyön aikana tehtiin Android-mobiilikäyttöjärjestelmälle tehdystä widget-ohjelmasta käännös Symbian^3-käyttöjärjestelmälle. Opinnäytetyön aikana tehtiin käännöksen suunnittelu, toteutus sekä yksikkötestaus. Laajempi testaus suoritetaan opinnäytetyön jälkeisenä aikana, minkä jälkeen korjataan testausvaiheessa löytyneet virheet ennen sovelluksen julkaisua.

Tehdessäni tätä opinnäytetyötä olen tutustunut moniin uusiin sovellusalueisiin sekä kehitustekniikoihin ja työkaluihin. Työn alkuvaiheessa perehdyin Android-käyttöjärjestelmään, jotta käännettävän widget-sovelluksen toiminnallisuus selviäisi. Tämän jälkeen keräämieni tietojen perusteella aloin perehtyä Symbian^3-järjestelmään. Tämän selvitystyön aikana kävin läpi mahdolliset tekniikat, joilla sovellusten tekeminen kyseiselle järjestelmälle on mahdollista, ja keskityin erityisesti widget-sovellusten kehittämiseen. Opinnäytetyön aikana minulle muodostui selkeä kuva siitä, mitä vaiheita erityisesti Symbian-käyttöjärjestelmille tehtävät sovellusprojektit pitävät sisällään.

Widgetin kehittämisprosessi sisälsi monia eri sovelluskehityksen osa-alueita. Suunniteltaessa WRT-sovelluksen käyttöliittymää tuli ottaa huomioon mahdolliset vaihtelevat päätelaitteet, joilla sovellusta käytetään. Tutustuin projektissa myös hajautetun ohjelmoinnin periaatteisiin luodessani kahta itsenäistä sovellusta, jotka kommunikoivat keskenään. Vaikka kyseessä oli päätelaitteen tasolla tapahtunut hajautus, antoi se hyvän kuvan hajautuksen eduista ja haasteista.

Opinnäytetyön loppupuolella tavoitteena oli luoda kolmas sovellus, joka toimisi kahden kehitetyn sovelluksen asennusohjelmana. Näin testausvaiheessa sovelluksen jakelu olisi ollut yksinkertaisempaa, koska jaettavia sovelluspaketteja olisi ollut kahden sijasta yksi. Asennussovelluksen kanssa ilmeni kuitenkin ongelmia allekirjoitusvaiheessa, koska asennuspaketti vaati toimiakseen sertifioinnin Symbian Signed -palvelusta. Testausvaiheeseen pääsyn nopeuttamiseksi päätettiin asennuspakettiohjelma hylätä ja jakaa sovellus kahtena asennuspakettina.

Ohjelmointityö oli mielenkiintoista ja palkitsevaa, koska projektin aikana piti opetella ennestään tuntemattomat Qt- ja WRT-ohjelmointiympäristöt. Vaikka kumpikin näistä pohjautuu vanhempiin ohjelmointikieliin, oli niissä silti paljon uutta opittavaa.

Opinnäytetyön tekeminen selkeytti myös kuvaa nykyisestä mobiilien järjestelmien markkinatilanteesta. Android- ja Symbian^{^3}-järjestelmien hyvät ja huonot puolet tulivat hyvin ilmi, joten jatkossa sovellusten kehittäminen eri tarpeisiin eri järjestelmille on helpompaa.

Sekä Android- että Symbian-käyttöjärjestelmillä on sovelluskehityksen kannalta hyvät ja huonot puolet. Androidin selkeänä etuna on sen avoin ympäristö, joka helpottaa sovellusten kehittämistä etenkin pienempien sovelluskehitysyriyten osalta. Symbian-järjestelmän etu on Androidia pienempi käyttöjärjestelmäversioiden määrä, minkä ansiosta sovellusten suunnitteluvaiheessa ei tarvitse ottaa huomioon monia erilaisia päätelaitteiden ja käyttöjärjestelmäversioiden yhdistelmiä.

Opinnäytetyöprojekti oli kokonaisuudessaan mielenkiintoinen ja motivoiva. Projektin aikana ilmenneet ongelmat niin itse sovelluksen suunnittelussa kuin käytettyjen tekniikoiden käytössä kehittivät ongelmanratkaisukykyä ja kärsivällisyyttä. Henkilökohtaisena tavoitteena olleet Qt-ohjelmointiympäristön opettelu sekä perehtyminen mobiiliohjelmointiin käytännön tasolla saavutettiin.

LÄHTEET

Forum Nokia Wiki. *Symbian OS*. [viitattu 7.3.2011] Saatavissa:
http://wiki.forum.nokia.com/index.php/Symbian_OS

Forum Nokia Wiki. *Nokia WRTkit OverView*. [viitattu 7.3.2011] Saatavissa:
http://wiki.forum.nokia.com/index.php/WRTkit_Overview

Forum Nokia Wiki. *Create your first WRT widget*. [viitattu 10.3.2011] Saatavissa:
http://wiki.forum.nokia.com/index.php/Create_your_first_WRT_widget

SQLite. *About SQLite*. [viitattu 10.3.2011] Saatavissa: <http://www.sqlite.org/about.html>

Nokia Oyj. *Qt*. [viitattu 10.3.2011] Saatavissa: <http://qt.nokia.com/>

Open Handset Alliance. *Android Overview*. [viitattu 11.3.2011, 15.3.2011] Saatavissa:
http://www.openhandsetalliance.com/android_overview.html

Open Handset Alliance. *Overview*. [viitattu 11.3.2011] Saatavissa:
http://www.openhandsetalliance.com/oha_overview.html

Androidsuomi Wiki. *Android*. [viitattu 11.3.2011] Saatavissa:
<http://wiki.androidsuomi.fi/Android>