

Heikki Keskikallio

**TUOTANNONSUUNNITTELU- JA KAPASITEETINLASKENTASOVELLUKSEN  
SIIRTO S60 5th EDITIONILLE**

**TUOTANNONSUUNNITTELU- JA KAPASITEETINLASKENTASOVELLUKSEN  
SIIRTO S60 5th EDITIONILLE**

Heikki Keskikallio  
Opinnäytetyö  
Syksy 2010  
Tietotekniikan koulutusohjelma  
Oulun seudun ammattikorkeakoulu

Koulutusohjelma	Opinnäytetyö	Sivuja	+	Liitteitä
Tietotekniikan koulutusohjelma	Insinöörityö	50	+	0
Suuntautumisvaihtoehto	Aika			
Sulautetut järjestelmät	2010			
Työn tilaaja	Työn tekijä			
Qprojects Oy	Heikki Keskkikallio			
Työn nimi				
Tuotannosuunnittelu- ja kapasiteetinlaskentasovelluksen siirto S60 5 <sup>th</sup> editionille				
Avainsanat				
Symbian S60, porttaaminen, ERP, Qt				

Nykyäänä yritykset, jotka toimivat tuotantosektorilla, eivät tulisi toimeen ilman toimivaa ratkaisua tuotannonohjaukseen. Tuotannonohjaukseen tarvitaan, jotta yritys voi toimia tehokkaasti eli vastata mahdollisiin ongelmiin nopeasti ja minimoida menetykset, joita ongelmat tuotannossa mahdollisesti aiheuttavat. Tätä varten on kehitetty erilaisia tietokonepohjaisia ERP-ratkaisuja, joiden avulla yritykset voivat vastata tähän ongelmaan.

Opinnäytetyössä portattiin aiemmin kehitetty mobiili jatke työpöytäympäristössä ajettavalle tuotannosuunnittelu- ja kapasiteetinlaskentasovellukselle. Porttaus suoritettiin nykyisille Nokian kosketusnäyttöpuhelimille, jotka tukevat S60 5th Edition -käyttöliittymää. Sovellukseen lisättiin myös uusia ominaisuuksia. Opinnäytetyössä otettiin lyhyesti myös selvää, millä tavalla Qt-ohjelmointia voitaisiin hyödyntää sovelluksen jatkokehittämisessä.

Lopputuloksena syntyi toimiva porttaus alkuperäisestä ympäristöstä (Series 80) Nokian S60 5th Editionia hyödyntäville kosketusnäyttöpuhelimille. Sovellus mahdollistaa pääsyn samaan tietoon, joka on käytössä myös sovelluksen työpöytäversiossa. Sovellus on edelleen jatkuvan kehityksen kohteena ja uusia toiminnallisuuksia on kehitteillä tässä opinnäytetyössä lisättyjen jatkoksi.

Degree programme	Thesis	Number of pages	+ appendices
Information Technology and Telecommunications	B.Sc.	50	+ 0
Line	Date		
Embedded Systems	2010		
Commissioned by	Author		
Qprojects Oy	Heikki Keskikallio		
Thesis title	Porting of manufacturing and capacity calculation application to S60 5 <sup>th</sup> edition		
Keywords	Symbian S60, porting, ERP, Qt		

In modern days companies that operate in production sector wouldn't be able function efficiently without a solution for production management. Production management is needed so that companies can work efficiently, that is to answer to the possible issues caused by problems in production. For this it has been developed different kinds of computer based ERP-solutions to answer to this dilemma.

In this thesis, a mobile extension to an already existing manufacturing and capacity calculation application was ported to current Nokia touch enabled smartphones using S60 5<sup>th</sup> edition as their user interface. New features were also added to the application. In addition to those, it was also briefly studied how Qt-programming could be made use of to enhance the application in future development.

As a result of this thesis, a working port from the original environment (Series 80) to touch enabled smartphones utilizing Nokia S60 5<sup>th</sup> edition was received. The application enables access to the same information that is at hand in the desktop version. The application is still under development and new features are planned in addition to those that were added in this thesis.

# SISÄLTÖ

TIIVISTELMÄ .....	3
ABSTRACT .....	4
SISÄLTÖ.....	5
LYHENTEET JA NIIDEN MERKITYKSET .....	7
1 JOHDANTO .....	9
2 M4MAMI .....	10
2.1 ERP .....	10
2.2 MAMI ja M4MAMI.....	11
2.2.1 Middleware.....	13
2.2.2 Tietokannat .....	13
2.2.3 Käyttöliittymä.....	14
3 SYMBIAN OS.....	18
3.1 Historia .....	18
3.2 Symbianin tulevaisuuden näkymät.....	19
3.3 Symbian OS:n ominaisuudet.....	20
3.3.1 Kernel.....	21
3.3.2 Muistinhallinta .....	21
4 SYMBIAN C++ .....	23
4.1 Hakemistorakenne ja tiedostotyypit.....	23
4.2 Sovellusten luokkarakenne .....	24
4.3 Oliot ja virnehallinta .....	25
5 KEHITYSYMPÄRISTÖN PYSTYTTÄMINEN.....	27
5.1 Tarvittavat komponentit .....	27
5.2 Ilmenneet ongelmakohdat.....	28
6 TYÖN SUORITUS .....	30
6.1 S60 5th Editionille porttaus.....	30
6.2 Vaiheet .....	33
6.3 Muutokset tiedostoihin.....	33
6.3.1 mmp-tiedostot .....	34
6.3.2 Lähdekooditiedostot.....	36
6.3.3 Resurssitiedostot.....	37
6.3.4 Sovelluksen paketoititiedosto.....	39
6.3.5 Kosketusnäyttötuen lisääminen .....	39

7 JATKOKEHITYS.....	43
7.1 Qt:n alustariippumattomuus .....	44
7.2 Qt:n hyödyntäminen .....	44
8 POHDINTA .....	46
LÄHTEET.....	48

## LYHENTEET JA NIIDEN MERKITYKSET

C++	Ohjelmointikieli
CPU	Central Processing Unit – prosessori
DMA	Direct Memory Access – oikosiirto, tiedon kopiointi laitteen sisällä kuljettamatta sitä suorittimen kautta
GCM	Generic Communications Middleware
GUI	Graphical User Interface – graafinen käyttöliittymä
DBMS	DataBase Management System – tietokannan hallintajärjestelmä
EKA2	Symbian OS kernel architecture – kernelin arkkitehtuuri
ERP	Enterprise Resource Planning – toiminnanohjausjärjestelmä
IDE	Integrated Development Environment – graafinen sovelluskehitysympäristö
JRE	Java Run-time Environment
M4MAMI	Mobile for MAMI
MAMI	Mass production Administration by Multiple Inquiries
MMU	Memory Management Unit – muistinhallintayksikkö
MRP II	Manufacturing Resource Planning
MVC	Model View Controller – suunnittelumalli

OS	Operating System – käyttöjärjestelmä
QML	Qt Meta-Object Language – käyttöliittymien kuvauskieli
Qt	Graafinen käyttöliittymäkirjasto
RAM	Random Access Memory – luku- ja kirjoitusmuisti
ROM	Read Only Memory – lukumuisti
SAP	Systems, Applications and Products
S60	Symbian OS:n päällä käytettävä käyttöliittymä
SDK	Software Development Kit – sovelluskehitystyökalut
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UI	User Interface – käyttöliittymä
UID	Unique Identifier – yksilöllinen tunniste



# 1 JOHDANTO

Tuotannonohjaus on merkittävä osa monien yritysten liiketoimintaa. Yrityksen tuotannon täytyy pystyä sopeutumaan niihin ehtoihin, joita tilattujen tuotteiden volyyymi sille aiheuttaa. Tuotannonohjaukseen kuuluvia asioita ovat muun muassa tuotannon ajoitus, varastojen valvonta sekä tuotantokapasiteetin tehokas hyödyntäminen. Ilman toimivaa tuotannonohjausta yritys ei voi toimia kannattavasti voimakkaasti kilpailluilla markkinoilla.

Työ toteutettiin Qprojects Oy:lle, joka sijaitsee Haukiputaalla. Yritys on perustettu 1990 ja toimii ohjelmistoalalla. Yritys tuottaa vaativia ohjelmistoalan ratkaisuja pääasiassa teollisuuden tarpeisiin ja sen tuotteita käytetään päivittäin osana merkittävien kansainvälisten toimijoiden liiketoimintaa.

Opinnäytetyön tavoitteena oli portata jo olemassa oleva tuotannosuunnittelu- ja kapasiteetinlaskentasovellusta (MAMI, Mass production Administration by Multiple Inquiries) hyödyntävä mobiili jatke Symbian S80 -alustaa käyttävältä Nokia 9500 Communicatorilta nykyisille Nokian puhelimille, jotka käyttävät Symbian S60 5th Edition -alustaa. Tarkoituksena oli selvittää se prosessi, jonka porttaustyö vaatii. Porttaamisella tarkoitetaan tässä sitä vaihetta, kun sovellus siirretään alkuperäistä ympäristöstä toiseen, jossa se ei toimi suoraan, vaan vaatii muutoksia ohjelmakoodiin. Pääpaino oli uusien puhelimen tukeman kosketusnäyttöominaisuuden hyödyntämisessä. Porttaamisen yhteydessä otettiin myös huomioon ne vaatimukset, jotka ovat ominaisia kosketusnäyttöominaisuuksia käyttäville sovelluksille. Porttaamisen lisäksi sovellukseen toteutettiin ominaisuus, jonka avulla voidaan selvittää, liittyykö sovelluksen esittämiin tilauksiin materiaali- tai kapasiteettipuutteita. Omakohtaisena tutkimuksena otin mukaan Qt-ohjelmoinnin, jota käsittelen tarkemmin luvussa 7.

## 2 M4MAMI

M4MAMI (Mobile for MAMI) on Qprojects Oy:n ja VTT:n yhteistyöprojekti, joka sai alkunsa vuonna 2005. Tavoitteena oli kehittää mobiili jatke olemassa olevaan tuotannosuunnittelu- ja kapasiteetinlaskentasovellukseen, MAMlin. M4MAMI mahdollistaa saman tiedon esittämisen mobiilisti, joka on nähtävillä MAMlin työpöytä-clientissa. Sovellus kehitettiin alunperin Nokia 9500 Communicatorille. (15.)

M4MAMI hyödyntää Symbianin DBMS:ää (Data Base Management System). Sovellus tallentaa selatun tiedon omaan tietokantaansa, jotta siihen päästään käsiksi myös silloin, kun verkkoon ei ole pääsyä. Tietokanta on uudelleen suunniteltu ja yksinkertaistettu mobiiliversiota varten. Toinen mahdollisuus on synkronoida haluttu tieto tietyltä ajalta sovelluksen tietokantaan offline-selausta varten.

Tässä osiossa taustoitetaan lyhyesti M4MAMlin toimintaympäristöä sekä tutustutaan lähemmin MAMlin ja M4MAMlin. Tarkoitus ei ole kuitenkaan esitellä MAMla kokonaisuudessaan vaan lyhyesti kertoa, mikä MAMI on ja mihin se liittyy.

### 2.1 ERP

ERP (Enterprise Resource Planning) tarkoittaa tuotannonohjausjärjestelmää. ERP liittyy olennaisesti MAMlin. Se on järjestelmä, joka kokoaa ja hallinnoi yrityksen sisäisiä ja ulkoisia resursseja yhtenä kokonaisuutena. Järjestelmän avulla yritykset voivat tehostaa omaa toimintaansa. Esimerkiksi yrityksen eri osastot pystyvät helpommin seuraamaan toistensa tekemisiä ja näin välttämään päällekkäisten projektien syntymistä. (3, hakusana Enterprise Resource Planning.)

ERP tuli terminä esille 1990-luvun alussa. Tällöin kehitys- ja konsultointiyritys Gartner Group (nykyisin Gartner) esitteli sen jatkeena MRP II -

järjestelmälle (Manufacturing Resource Planning), joka oli tarkoitettu tuotantopainotteisille yrityksille tavaksi hoitaa yrityksen resurssien tehokas käyttö. MRP II -järjestelmä on siis osa ERP-järjestelmää, jonka tavoitteena on pystyä vastaamaan kaikkien yritysten tarpeita, riippumatta siitä, mitkä ovat yrityksen ydintoimintoja. (3, hakusana Enterprise Resource Planning.)

ERP-järjestelmille tunnuksenomaisia piirteitä ovat seuraavat asiat: sen tulee olla integroitu ja olla tosiaikainen. Kaikkien siihen liittyvien sovellusten tulee liittyä yhteen tietokantaan (ERP-järjestelmän tietokanta), jotta vältettäisiin saman data määrittely useassa eri paikassa. Sen sisäisten moduulien tulee olla yhtenäisiä toistensa suhteen ja käyttäjillä tulisi olla pääsy sen sisältämään tietoon ilman ylimääräistä vaivannäköä. (3, hakusana Enterprise Resource Planning.)

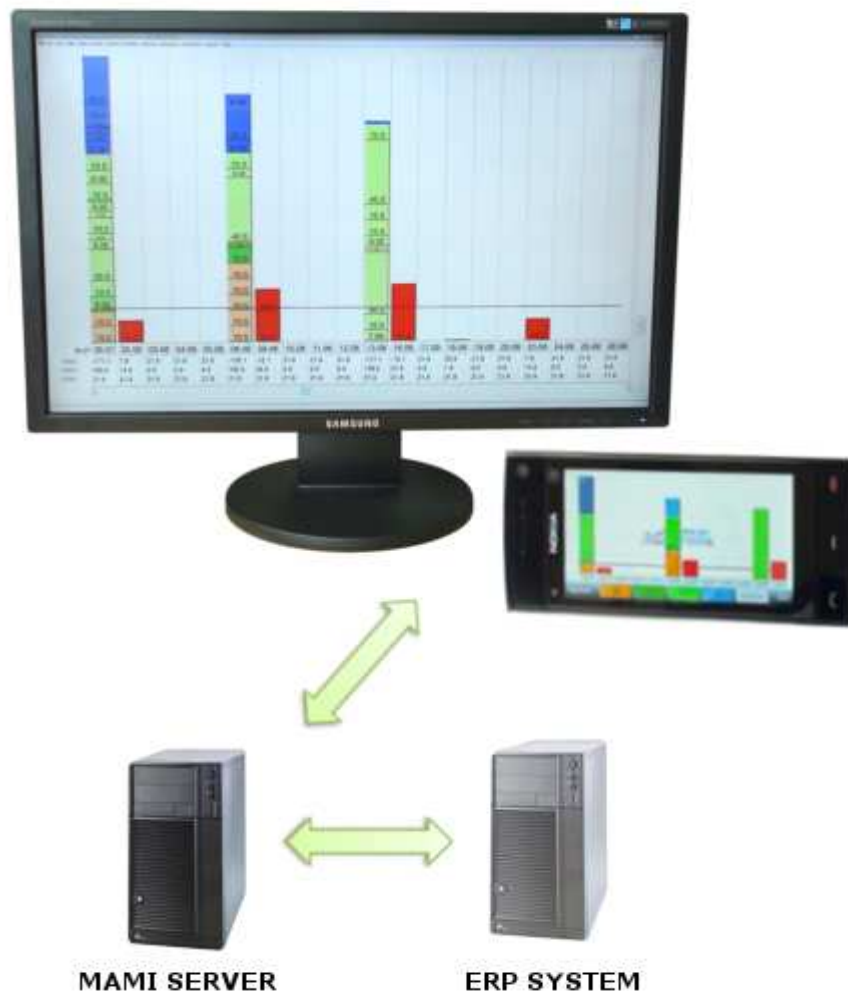
## **2.2 MAMI ja M4MAMI**

Alkuperäinen MAMI on Qprojects Oy:n vuonna 1995 kehittämä tuotannosuunnittelu- ja kapasiteetinlaskentasovellus informaatioteknologia-alan yrityksille. Sovellus on ollut siitä lähtien jatkuvan kehityksen alla, jotta se vastaa tämän päivän tarpeita. (19.)

MAMI yhdistää suuren osan yrityksen liiketoimintaan liittyviä prosesseja yhdeksi kokonaisuudeksi. Se valvoo yrityksen koko tuotantoa ja käsittelee useiden eri osastojen tuotteita. Tuotannosuunnittelusta vastaavat henkilöt voivat optimoida kapasiteettia, jotta tuotanto täyttäisi sille asetetut vaatimukset. (19.)

MAMIn työpöytä-clientin avulla pystytään tarkastelemaan sekä muokkaamaan tietokannassa olevaa tietoa. MAMI-clientin esittämä tieto on haettu ERP-järjestelmästä MAMI-serverin tietokantaan. Tietokanta koostuu muun muassa myyntitilauksista, ennusteista, sisäisistä ja ulkoisista ostotilauksista, nimiketiedoista ja tuoterakenteista. MAMIn käyttäjiä ovat muun muassa myyntihenkilökunta, tuotannosuunnittelusta vastaavat sekä yrityksen johto. Jokainen heistä käyttää MAMIlta omaan tarkoitukseensa. Myynnistä vastaa-

vat käyttävät sitä kapasiteetin tarkkailuun pitääkseen kiinni tuotteille asetetuista toimituspäivistä. Tuotannosuunnittelijat puolestaan käyttävät sitä kapasiteetin laskuun siten, että tuotanto on riittävää. Johto tarkkailee sen avulla osastojen tuotantoa sekä kapasiteettia. Kuvassa 1 on esitetty MAMI-järjestelmä, jossa on myös tämän lopputyön tuloksena syntynyt porttaus M4MAMIsta S60 5th Editionia hyödyntäville älypuhelimille. M4MAMI tukee edelleen myös S60 3rd Edition -puhelimia, ennen kaikkea Nokia E90 Communicatoria. Kun opinnäytetyötä tehtiin, pidettiin yllä jatkuvasti tukea myös kyseiselle kommunikaattorille.



*KUVA 1. MAMI-järjestelmä*

## 2.2.1 Middleware

Middlewareella tarkoitetaan ohjelmistoa, jonka avulla voidaan yhdistää useampi ohjelmistokomponentti toimimaan yhdessä. M4MAMlissa tämä ohjelmisto toimii väylänä, jonka avulla M4MAMI kommunikoii MAMI-serverin kanssa välityspalvelimen (proxy) kautta. Lähetettäviä GCM-viestejä (Generic Communications Middleware) varten on kehitetty erityinen GCM-protokolla tiedonsiirtoa varten. GCM-protokolla takaa luotettavan tiedonsiirron yhteysosapuolten välillä. Mikäli GCM-viestien lähettäminen ei onnistu, tulee vastaanottavaa osapuolta informoida tapahtumasta. (3, hakusana Middleware; 16.)

Protokolla on suunniteltu hyvin helppokäyttöiseksi tavaksi keskustella clientin ja serverin välillä. Protokolla on riippumaton tiedonsiirron tavasta. Se ei itessään välitä siitä, kuinka monta parametria viestissä välitetään. Täten jos halutaan lisätä uusia toiminnallisuuksia sovellukseen, ei protokollaan tarvi tehdä mitään muutoksia. Sovelluskehittäjän täytyy ainoastaan huolehtia siitä, millä nimellä parametreja lähetetään ja vastaanotetaan. Serverin ja clientin tehtävänä on tutkia, mitä on vastaanotettu, ja toteuttaa niitä vastaava toiminnallisuus. Erilaisia toiminnallisuuksia ovat muun muassa tilausrivien lähettäminen, materiaali- ja kapasiteettipuutetietojen lähettäminen, synkronoinnissa käytettävien osastojen lisäys tai poisto sekä synkronointipyyntö.

## 2.2.2 Tietokannat

Tietokanta on kokoelma toisiinsa liittyvää dataa, joka on tallennettu suunnitellusti tietokoneelle. Sieltä sitä voidaan hakea erilaisin kyselyin ja tehdä siihen muutoksia. (3, hakusana Tietokanta.)

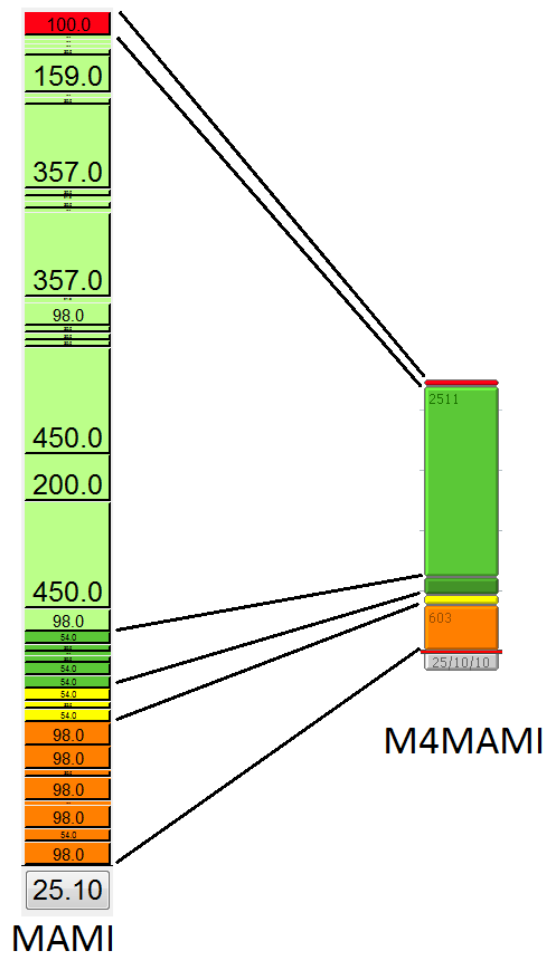
Tietokannat ovat olennainen osa lähes kaikkialla missä tietokoneita käytetään. Ilman niitä monet sovellukset eivät voisi toimia tarkoituksenmukaisella tavalla. ERP-järjestelmät ovat hyvä esimerkki tästä. Niihin kuuluu olennaisena osana tietokanta, jossa on monenlaista tietoa yrityksen tarpeisiin. MAMI käyttää tätä tietokantaa, kun se hakee SAP-järjestelmästä (Systems, Appli-

cations and Products) tarvittavat tiedot omaan tietokantaansa. M4MAMI sisältää oman tietokantansa, jonka hallintaan käytetään Symbian DBMS:ää. Se aiheuttaa omat rajoituksensa M4MAMIn tietokannalle, sillä kyseinen tietokantajärjestelmä ei sisällä kaikkia niitä ominaisuuksia, mitä muut järjestelmät tarjoavat. Symbian DBMS ei esimerkiksi tue taulujen yhdistämistä, mistä aiheutuu redundanssia, eli samaa tietoa joudutaan tallentamaan useaan eri tauluun. Tästä johtuen tietokantaa ei ole voitu suunnitella yhtä tehokkaaksi, kuin se on MAMIn puolella. (3, hakusana Tietokannan normalisointi.)

Tietokantojen oikeanlainen suunnittelu ja käyttö on olennaisen tärkeää, kun puhutaan järjestelmistä, joille niiden oikeanlainen toiminta on elinehto. M4MAMIn porttauksen yhteydessä huomattiin, että synkronoinnissa käytettyä hakua serveriltä voitiin nopeuttaa huomattavasti toteuttamalla haku uudella tavalla. Tämä moninkertaisti tietojen synkronoinnin nopeuden.

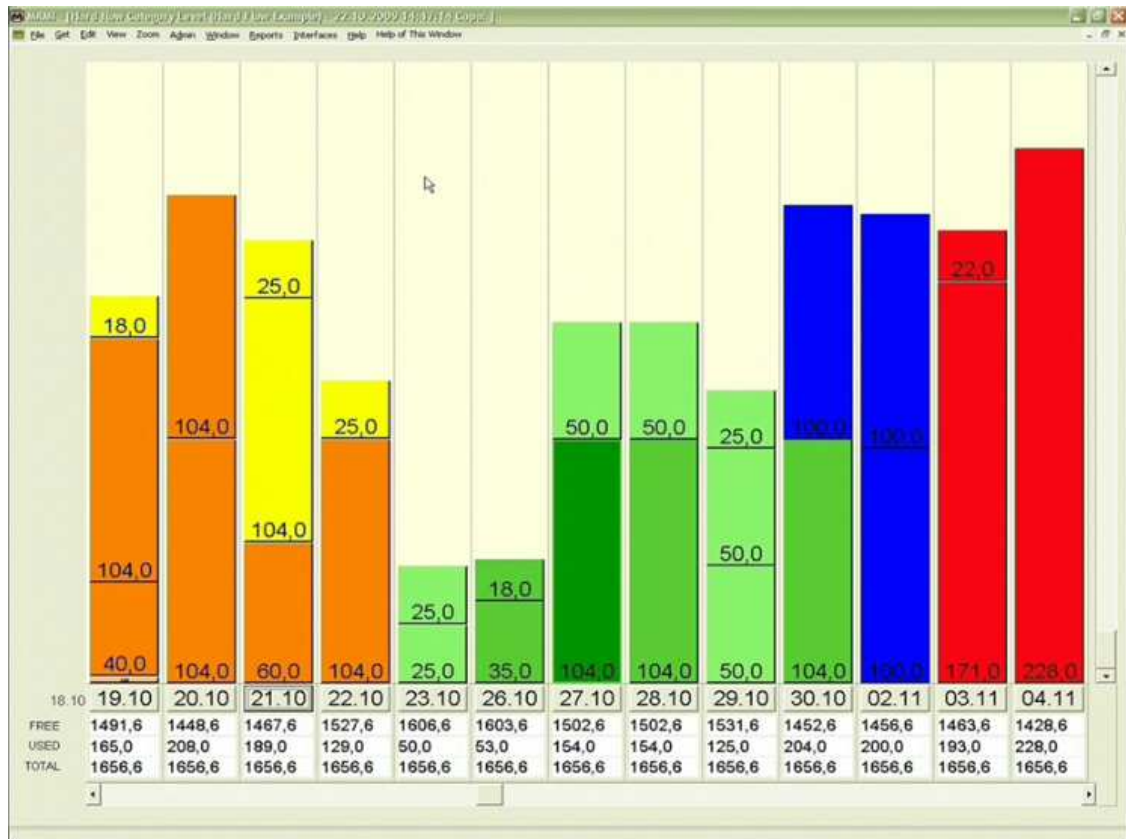
### **2.2.3 Käyttöliittymä**

MAMIn käyttöliittymä esittää MAMI-serveriltä haetut tiedot graafisesti, ja ne on jaoteltu statuksen mukaan eri väreillä. Väreillä jaotellut lohkot sisältävät tietoa tilauksista. Oranssi väri tarkoittaa toimitettua tilausta, keltainen työn alla olevaa tilausta, vaalean vihreä osittain vahvistettua tilausta – tämä tarkoittaa sitä, että lohko pitää sisällään sekä vahvistettuja että vahvistamattomia tilausrivejä –, vihreä vahvistettuja tilauksia – tämä puolestaan sitä, että tuotanto tai toimitus voidaan toteuttaa, kuten on suunniteltu –, sininen vahvistamattomia tilauksia ja punainen tarkoittaa joko vahvistettua tilaussuunnitelmaa tai ennustetta. M4MAMlssa samalla statuksella olevat tilaukset on niputettu yhteen toisin kuin työpöytäversiossa, kuten kuva 2 osoittaa.



*KUVA 2. MAMIn ja M4MAMIn lohkonäytön eroavaisuudet*

Tilausten niputus on tehty helpottamaan luettavuutta sekä käytettävyyttä, sillä tilauksia voi olla satoja yhdellä pylväällä. Tilauksien esittäminen kärsisi, mikäli ne pyrittäisiin esittämään kuten työpöytäversiossa. Tilaukset voidaan esittää halutulla tarkkuudella tietyltä aikaväliltä. Kuvassa 3 on esitetty MAMIn työpöytä-clientin käyttöliittymä.



KUVA 3. MAMI-clientin käyttöliittymä

Avaamalla lohko M4MAMlissa saadaan esille tietyllä statuksella olevat tilaukset. Tilaukset esitetään listana. Niihin voi liittyä materiaali- tai kapasiteetti- puutteita, jotka ovat tärkeitä asioita tuotannonohjauksen kannalta. MAMlissa puutteet voidaan esittää visuaalisesti suoraan lohkonäkymässä. Puutteelliset lohkot esitetään niin, että niiden päälle piirretään ristikko.

M4MAMlin lisättiin osana tätä opinnäytetyötä mahdollisuus tarkastella tilauksiin liittyviä puutteita, joka puuttui aiemmasta versiosta. Puutteita voidaan tarkastella M4MAMlissa usealla eri tarkkuudella. Puutteet saadaan näkyviin avaamalla lohko, joka kertoo, mihin tilauksiin puutteita liittyy, esittämällä kyseinen tilausrivi punaisella värillä. Puutteita voidaan ajatella puurakenteena, kun ajatellaan tietyn tilauksen koostuvan useasta eri komponentista, joka puolestaan koostuu muista komponenteista, joihin voi liittyä puutteita. Täten puutteet muodostavat puurakenteen tuotantoketjussa. Määrittämällä taso puutteita haettaessa voidaan määrittellä, kuinka syvältä puurakenteesta niitä haetaan.



Materiaali- ja kapasiteettipuutetietojen merkitys tuotannosuunnittelussa on erityisen tärkeää. Mikäli nämä eivät kohtaa, koituu siitä välittömiä menetyksiä yrityksen liiketoiminnalle, kuten toimitusten myöhästymiset ja materiaalien liian aikainen saapuminen. Sen takia on välttämätöntä, että tuotantoa pystytään sopeuttamaan vallitsevaan tilanteeseen sopivalla tavalla, jotta voidaan minimoida aiheutuvia menetyksiä. Tällä taataan hyvä toimitustarkkuus ja varastojen kiertonopeus sekä minimoidaan keskeneräisen tuotannon arvo.

Käytännössä M4MAMIn avulla päästään käsiksi samaan tietoon, joka on käytettävissä myös MAMI-työasemaversiollakin. M4MAMI on erittäin helppokäyttöinen ja toimintavarma sekä liikuteltavuutensa ansiosta aina käytettävissä, riippumatta siitä, missä myyntihenkilö puhelimeen liikkuu. M4MAMIn ansiosta myyntihenkilö kykenee helposti selvittämään kunkin asiakkaan olemassa olevien tilausten tilan reaaliaikaisesti sekä vahvistamaan mahdollisten asiakkaiden lisätilausten toimitusajankohdan suoraan tehtaan tilauskuormaa vastaten. M4MAMIlta ei ole tarkoitettu korvaamaan työpöytäversiota, vaan olemaan hyödyllinen apuväline silloin, kun MAMIn sisältämään tietoon ei ole muuta pääsyä kuin matkapuhelin.

## 3 SYMBIAN OS

Symbian OS (Symbian Operating System) on Nokian omistama käyttöjärjestelmä älypuhelimille. Se on tämän hetken käytetyin käyttöjärjestelmä älypuhelimissa. Muut laitevalmistajat ovat kuitenkin viime vuosina siirtyneet käyttämään joko itsensä tai muiden kehittämiä käyttöjärjestelmiä. Esimerkiksi Samsung ja Sony Ericsson ovat lopettamassa tukeaan Symbianille. (1; 2.)

### 3.1 Historia

Alunperin Symbiania kehitti Symbian Ltd. Symbian pohjautuu Psionin kehittämään EPOC-käyttöjärjestelmään, joka toimii yksinomaan ARM-proessoreilla. Jotta saataisiin käsitys siitä, kuinka pitkän työn tulos Symbian OS kaikkiaan on, täytyy lähteä liikkeelle vuodesta 1980, jolloin David Potter perusti Psionin. Psion ehti kehittää vuonna 1984 maailman ensimmäisen kädessä pidettävän tietokoneen (Psion Organiser), joka muistutti lähinnä sen aikaisia pelilaitteita. Vuonna 1987 Psion alkoi kehittää uusia 16-bittisiä Organiser-laitteita ja niillä toimivaa EPOC-käyttöjärjestelmää, myöhemmin EPOC16. (3, hakusana Symbian OS.)

32-bittisen EPOCin (EPOC32) kehitys aloitettiin vuonna 1994 ja se julkaistiin 1997. Vuosi julkaisun jälkeen, kesäkuussa 1998, EPOCia käyttävät matkapuhelinvalmistajat perustivat Symbian Ltd:n, johon kuuluivat Psion, Nokia, Ericsson ja Motorola. EPOC32 release 6 -version myötä käyttöjärjestelmän nimeksi tuli Symbian OS. (3, hakusana Symbian OS.)

Symbian Ltd. ehti toimia seuraavat 10 vuotta, kunnes Nokia ilmoitti hankkivansa koko Symbian Ltd:n omistukseensa. Kauppa valmistui joulukuussa 2008. Samana vuonna Nokia ilmoitti perustavansa voittoa tavoittelemattoman Symbian Foundationin jatkamaan Symbianin kehitystyötä. (4.)

## 3.2 Symbianin tulevaisuuden näkymät

Nokia on ilmoittanut luopuvansa nykyisestä tavasta nimetä Symbian-käyttöjärjestelmän eri versiot. Tulevaisuudessa Nokia kutsuu sitä yksinkertaisesti Symbianiksi, josta jätetään versionumerointi pois. Käyttöjärjestelmää kehitetään päivityksin, jotka tulevat käyttäjien ladattaviksi. Aiemmin Nokia kehitti Symbian ^4 -versiota, jonka erillinen kehitys on nyt lakkautettu. Versiolla oli kaavailtu pohjautuvan täysin Qt-ohjelmointiympäristöön, jossa perinteinen S60-käyttöliittymän käyttämä AVKON-käyttöliittymäkirjasto tultaisiin korvaamaan Qt-käyttöliittymäkirjastolla, jonka Nokia osti norjalaiselta Trolltechilta vuonna 2008. Tämä olisi tarkoittanut sitä, että vanhat sovellukset eivät enää jatkossa olisi yhteensopivia uuden Symbian ^4 -version kanssa, jossa sovellusten käyttöliittymien luontiin käytetään Qt-tekniikkaa (4). AVKON on siis edelleen mukana nykyisessä Symbian ^3 -versiossa. Nokia on kuitenkin samalla painottanut sitä, että Qt tulee olemaan heidän ainoa ohjelmointiympäristönsä tulevaisuudessa ja että kaikki sovellukset kehitetään Qt:llä niin Symbian- kuin Meego-laitteille. (14.)

Voidaan kuitenkin olettaa, että Symbian ^4 -version kehityksen parissa tehty työ tulee myös osaltaan näkymään päivityksinä nykyisessä Symbian ^3 -versiossa. Esimerkkinä ovat ne hyvät ideat, joita käyttöliittymän kehityksessä on saatu. Kuvassa 4 on esitetty varhainen visio siitä, miltä tulevan Symbian ^4 -version käyttöliittymän ajateltiin näyttävän.



KUVA 4. Symbian ^4 -version UI-asettelu (17.)

### 3.3 Symbian OS:n ominaisuudet

Symbian OS on kehitetty erityisesti mobiileja vähävirtaisia laitteita varten. Tämä on vaikuttanut merkittävästi käyttöjärjestelmän kehitykseen, sillä laitteen tulee kyetä toimimaan pitkiä aikoja yhdellä latauksella. Nykyajan puhelimiin tulee jatkuvasti uusia ominaisuuksia, jotka rasittavat laitteen akkua aiempaa enemmän. Lisäksi puhelimia käytetään usein myös pelaamiseen sekä musiikin ja videoiden toistoon, jotka ovat voimakkaasti virtaa kuluttavia toimintoja. Tästä johtuen käyttöjärjestelmään on lisätty virranhallintaominaisuuksia, jotka mahdollistavat puhelimen pidemmän käyttöajan. (5.)

Itse käyttöjärjestelmä sijaitsee laitteen ROM-muistilla (Read Only Memory), jossa sijaitsee myös käyttöjärjestelmään sisäänrakennettu middleware (7, s. 18) ja sovellukset. Järjestelmän ROM-muisti on mapattu asemaksi z:, ja siellä olevia sovelluksia yleensä myös ajetaan suoraan ROM-muistilta eikä niitä ladata ensin RAM-muistille (Random Access Memory). Puhelimen RAM-muistia käyttävät parhaillaan ajettavat sovellukset sekä itse käyttöjärjestelmä.

Tässä osiossa on kuvailtu lyhyesti eräitä asioita, jotka ovat liittyvät olennaisesti Symbian OS -käyttöjärjestelmään.

### **3.3.1 Kernel**

Symbian OS -käyttöjärjestelmän ytimenä toimii 32-bittinen keskeyttävään moniajioon kykenevä kernel, jossa on ominaisuuksia sekä mikrokernelistä sekä monoliittisestä kernelistä. Symbian OS:n kernelistä käytetään nimitystä EKA2. Kernelin tehtävänä on hallinnoida laitteistoresursseja, kuten RAM-muistia, ja tarjota sovelluksille pääsy näihin resursseihin. Hallinnoitavia laitteistoresursseja ovat muun muassa seuraavat: prosessori (CPU, Central Processing Unit) ja muistinhallintayksikkö (MMU, Memory Management Unit), muistinhallinnan käytänteet, keskeytysten hallinta ja DMA-kanavan (Direct Memory Access) hallinta. (7, s. 20.) EKA2:n ominaisuuksiin kuuluvat seuraavat asiat:

- Modulaarinen: Käyttöjärjestelmän toiminnallisuus on toteutettu erillisissä osissa yhden monoliittisen yksikön sijaan.
- Yksi käyttäjä: Symbian OS -käyttöjärjestelmällä voi olla ainoastaan yksi käyttäjä kerrallaan. Ei tue useita yhtäaikaista kirjautumisia.
- Moniajo: Kernel jakaa prosessorin aikaa useiden säikeiden kesken antaen vaikutelman siitä, että useat sovellukset käyvät samaan aikaan.
- Keskeyttävä moniajo: Kernel pitää sisällään skedulerin (vuorottelija), joka vuorottelee säikeitä niiden prioriteetin perusteella prosessorilla.
- Tosiaikainen: Palvelut suoritetaan tietyn ajan kuluessa. (6.)

### **3.3.2 Muistinhallinta**

Muistinhallinnasta kernelissä vastaa niin kutsuttu muistimalli. Malli on vastuussa muun muassa toimenpiteistä kontekstin vaihdon yhteydessä. Kon-

tekstin vaihdolla tarkoitetaan sitä tapahtumaa, kun siirrytään säikeestä toiseen, joka on puolestaan kernelin skedulerin vastuulla. Mikäli tapahtuu kontekstin vaihto, jossa siirrytään samalla myös prosessista toiseen, täytyy kernelissä olevan muistinhallintayksikön tehdä useita toimenpiteitä, muun muassa tyhjentää välimuisteja.

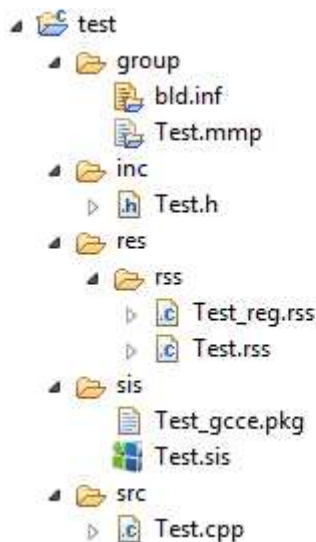
Jokainen käyttäjäpuolen sovellus Symbian OS:ssä on toteutettu prosessissa, jolla on vähintäänkin yksi säie, pääsäie. Tämä säie tuhoutuu, kun prosessi lopetetaan. Toimiakseen säie tarvitsee muistia, jota jokaiselle säikeelle varataan tietty määrä. Prosessille varattu muisti sijaitsee tietyssä virtuaalissa muistiavaruudessa, joten sovellusten raja on yhtä kuin prosessien välinen raja. Koska Symbian OS hyödyntää muistinsuojausta, ei prosessi voi kirjoittaa toisen prosessin varaamalle muistialueelle ja näin ollen vahingossa ylikirjoittaa toisen sovelluksen omistamaa dataa. Prosessin sisäiset säikeet voivat puolestaan kirjoittaa toisen säikeen varaaman muistin päälle, sillä ne sijaitsevat samassa muistiavaruudessa. (7, s. 23.)

## 4 SYMBIAN C++

Tässä luvussa on tarkoitus tutustua Symbian-sovellusten kehittämisessä huomioitaviin seikkoihin, joita ovat muassa sovelluksen rakenne, käytetyt tiedostotyypit sekä nimeämiskäytännöt.

### 4.1 Hakemistorakenne ja tiedostotyypit

Symbianissa on käytössä kuvan 5 mukainen hakemistorakenne. Siinä on esitetty sovelluskehityksen kannalta oleelliset kansiot sekä niiden sisältämät tiedostot.



KUVA 5. Symbian-sovelluksen hakemistorakenne

Tiedostotyyppien merkitykset:

- mmp-tiedosto: Projektin määrittelytiedosto, joka sisältää projektin tarvitsemat ominaisuudet. Tiedosto sisältää joukon käskyjä joiden perusteella generoidaan build-tiedostot emulaattorille sekä kohdelaitteelle.
- bld.inf-tiedosto: Komponenttien määrittelytiedosto, jota bldmake-työkalu käyttää määrittelemään abld.bat- ja make-tiedostot. Tiedosto koostuu eri osioista, kuten `prj_platforms`, `prj_exports`,

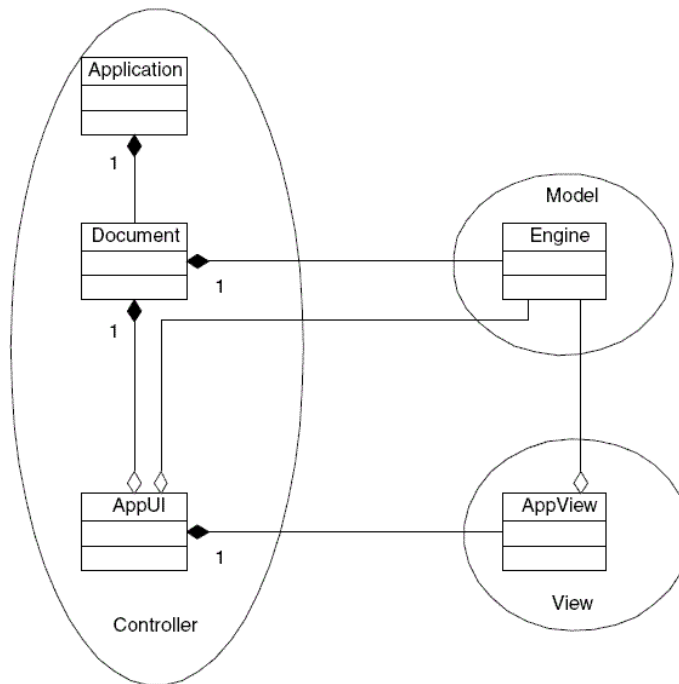
`prj_testexports` and `prj_mmpfiles`. Kaikkia näitä osioita ei välttämättä täydy olla `bld.inf`-tiedostossa, mikäli sovellus ei niitä tarvitse.

- `_reg.rss`-tiedosto: Sovelluksen rekisteröintiä varten oleva resurssitiedosto. Tämä tiedosto on ollut pakollinen Symbian v9.0:sta lähtien. Sen avulla rekisteröidään sovellus ja saadaan sovelluksen ikoni näkyväksi puhelimen sovellusmenuvalikkoon.
- `rss`-tiedosto: Resurssitiedosto, jossa on sovelluksen UI-kontrollien rakentamiseen tarvittava informaatio.
- `pkg`-tiedosto: Sovelluksen paketoitintiedosto, jonka avulla voidaan luoda sovelluksen asennustiedosto.
- `sis`-tiedosto: Sovelluksen asennustiedosto, jonka avulla sovellus voidaan asentaa kohdelaitteelle.
- `cpp`-tiedosto: Lähdekooditiedosto, joka sisältää sovelluksen luokkien toteutuksen.
- `h`-tiedosto: Otsikkotiedosto, joka pitää sisällään luokkien esittelyn.

## 4.2 Sovellusten luokkarakenne

Symbian tukee MVC-malliin (Model-View-Controller) perustuvaa ohjelmistokehitystä. Niinpä sovellusten luokkarakenne noudattaa Symbianille ominaista piirrettä, jossa luokat on jaoteltu MVC-mallin mukaisesti. Kuvassa 6 on esitetty Symbianissa käytettävä luokkien jaottelu.





KUVA 6. S60:n käyttämä MVC-malli (18.)

Model-View-Controller on suunnittelumalli (design pattern), jonka esitteli aikanaan Smalltalk-80. Kyseisessä mallissa Model vastaa engineä, joka pitää sisällään sovelluksen toimintalogiikan. View vastaa puolestaan sovelluksen näkymää ja Controller reagoi käyttäjän komentoihin. Viewin tulee heijastaa tilaa, jossa Model on. Tästä seuraa se, että sovellus voi tukea useita erilaisia näkymiä, joista kukin esittää mallin sisältämän informaation omalla tavallaan. Controller puolestaan ottaa vastaan syötteen (esimerkiksi näppäinpainallus), jonka perusteella Model suorittaa tietyn toimenpiteen, josta aiheutuu sovelluksen tilan muutos. Tämän seurauksena Model pyytää näytönpäivitystä. MVC-mallin käyttö helpottaa myös sovellusten porttaamista, sillä sovelluksen toimintalogiikkaan ei tarvitse tehdä muutoksia sen pysyessä samana alustasta riippumatta. (8, s. 4.)

### 4.3 Oliot ja virnehallinta

Olioiden rakentaminen ja tuhoaminen ovat tiiviisti sidoksissa keskenään. Koska puhelimia pidetään yleensä päällä hyvin pitkiä aikoja, on hyvin tärkeää, että olioiden rakentamisessa varattuja resursseja ei jää vapauttamatta eli ei tapahdu niin sanottua muistivuotoa. (7, s. 61.) Virheen tapahtuessa sovel-

luksen tulisi kyetä toipumaan siitä ja vapauttaa mahdollisesti varaamansa resurssit.

Symbianissa oliot, jotka on periytetty CBase-luokasta, tulee rakentaa kaksivaiheisen rakentamisen avulla. Kaksivaiheisessa rakentamisessa olioiden rakentaminen jaetaan kahteen eri vaiheeseen. CBase-luokasta periytyneen luokan tulee sisältää kaksi staattista NewL- ja NewLC-metodia, jotka ovat ensimmäisen vaiheen rakentajat. NewL-metodi on puolestaan toteutettu NewLC-metodin avulla, jonka sisältä kutsutaan kaksivaiheisen rakentamisen toista vaihetta, ConstructL-metodia. (7, s. 83.)

Kaksivaiheiseen rakentamiseen liittyy olennaisesti siivouspinon (CleanupStack) käyttö. Siivouspinoon asetetaan rakennettavan olion osoitin ennen kaksivaiheisen rakentamisen toista vaihetta. Mikäli olioiden rakentamisen toinen vaihe menee läpi ilman ongelmia, poistetaan olioiden osoitin siivouspinosta. Mikäli tapahtuu niin sanottu *Leave* eli standardia C++:aa vastaava *throw* rakentamisen toisessa vaiheessa, tuhotaan siivouspinoon sijoitetut oliot eikä muistivuotoa pääse tapahtumaan. Symbianin nimeämiskäytäntönä on lisätä *L*-kirjain jokaisen metodin perään, joissa voi tapahtua *Leave*. *Leave*-tapahtuma saadaan kiinni erityisen TRAP-mekanismien avulla, joka on olennainen osa virheenhallintaa Symbianissa. TRAP-mekanismien avulla voidaan tehdä tarvittavat toipumistoimet, jotka *Leave* aiheuttaa, ja palata aiempaan toimivaan tilaan. (7, s. 73.)

## 5 KEHITYSYMPÄRISTÖN PYSTYTTÄMINEN

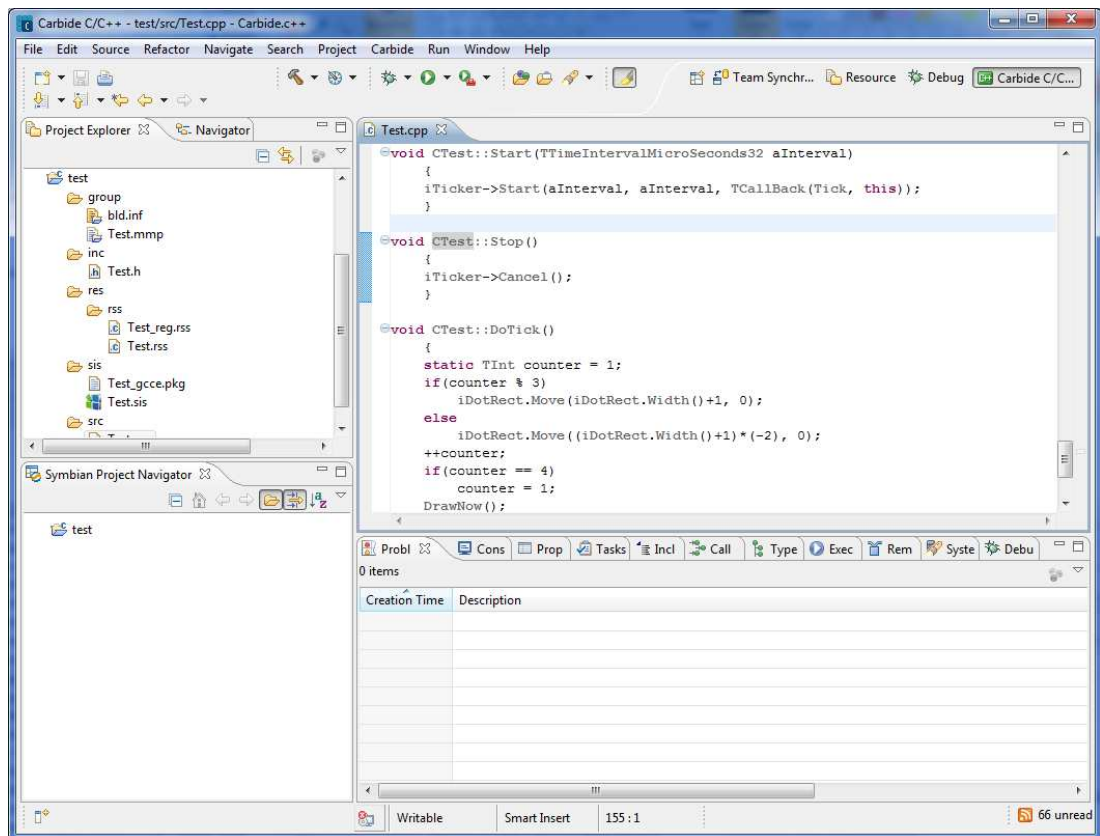
Jotta Symbian-sovelluksia voitaisiin kehittää, tarvitaan siihen sopivat työkalut. Nämä saadaan ladattua ilmaiseksi internetistä. Tässä luvussa esitellään sovelluskehitysympäristön pystyttäminen sekä siinä ilmenneet ongelmat ja niiden ratkaisuun vaadittavat toimenpiteet.

### 5.1 Tarvittavat komponentit

Symbian-sovelluskehityksessä tarvittavat komponentit ovat

- Carbide.c++ v2.x
- S60 SDK 5th Edition
- ActivePerl 5.6.1
- JRE.

Nokia tarjoaa nykyisin ilmaiseksi täyden version Carbide IDE:stä (Integrated Development Environment), joka on suositeltu kehitysympäristö Symbian-sovellusten rakentamiseen. Se tukee S60 3rd Editionia ja sitä uudempia SDK-versioita. Carbidella pystyy rakentamaan perinteisiä Symbian-sovelluksia sekä Qt-sovelluksia. Qt:stä kerrotaan lisää luvussa 7 (Jatkokehitys). Carbide on varsin kattava työkalu ja siinä on monia hyödyllisiä ominaisuuksia helpottamaan sovelluskehitystä. Carbiden käyttöliittymä on esitettyinä kuvassa 7.



KUVA 7. Carbide.c++ v2.3

Ennen SDK:n asennusta tulee asentaa ActivePerl 5.6.1. Myös uudemmat versiot toimivat nykyisin, sillä SDK:n asennusohjeen suosittelemaa ActivePerl-versiota ei ole enää saatavilla. ActivePerl saadaan ladattua ActiveStaten internet-sivuilta. Myös Java Run-Time Environment (JRE) 1.5.0 tai uudempi tulee olla asennettuna koneelle ennen SDK:ta. JRE löytyy nykyisellään Oraclen kotisivuilta. Oracle osti Javan kehittäneen Sun Microsystemsin vuonna 2009 (3, hakusana Sun Microsystems).

## 5.2 Ilmenneet ongelmakohdat

Windows Vista- sekä Windows 7 -versioiden kanssa ilmeni ongelmia kehitysympäristön toiminnassa. GCCE-kääntäjä ei toimi automaattisesti. S60 5th Edition SDK:n mukana tulee ohjeet, joilla kääntäjä saadaan toimimaan. Kehittäjän tulee suorittaa seuraavat vaiheet:

- Asenna S60 5th Edition SDK, mikäli ei ole jo asennettu.

- Kopioi SDK:n mukana tullut GCCE.mk-tiedosto hakemistoon C:\<SDK\_Installation\_dir>\epoc32\tools\compilation\_config.
- Kopioi SDK:n mukana tullut Env.BAT sovelluksen group-kansioon.
- Suorita Env.BAT-tiedosto group-kansiosta.
- Käännä sovellus GCCE-kääntäjällä kohdelaitteelle.

Vaihtoehtoisesti voidaan toimia myös niin, että Env.BAT-tiedostoa ei suoriteta. Env.BAT-tiedosto käytännössä asettaa ympäristömuuttajien PATH-muuttujan uusiksi. Voidaan siis avata Env.BAT editoitavaksi ja kopioida siitä tarvittavat polut ja asettaa ne PATH-muuttujaan. CSL ARM Tool Chain tulee olla asennettuna koneelle ennen tätä. S60 SDK:ta asennettaessa installer ilmoittaa mikäli sitä ei ole asennettu ja ehdottaa asennusta. Kehittäjän tulisi tässä vaiheessa hyväksyä asennuskehoitus.

Emulaattorin toiminnassa oli ongelmia myös samaisilla Windows-versioilla. Emulaattori kaatui lähes heti, kun sovellusta yritettiin debugata. Ongelmaksi paljastui äänikortti. Ongelma saatiin korjattua kytkemällä se pois päältä tietokoneen laitteiden hallinnan kautta. (9.)

## 6 TYÖN SUORITUS

Opinnäytetyön tavoitteena oli portata M4MAMI Nokia 9500 Communicatorilla, joka hyödyntää Symbian Series 80 -alustaa. Series 80 -alustasta käytetään myös nimeä Crystal. Tämä alusta kehitettiin alunperin ammattikäyttöön vastaamaan yritysten tarpeita. Etuina puhelimella oli aikanaan, kun M4MAMille valittiin sopivaa ympäristöä, muun muassa suuri näyttö, QWERTY-näppäimistö sekä tehokas prosessori ja muistin määrä. Lisäksi Series 80 hyödynsi SSL (Secure Sockets Layer)- sekä TLS-salausprotokollia (Transport Layer Security), joiden avulla pystyttiin salaamaan verkkoyhteyden yli siirrettävä data. (3, hakusana Series 80.)

### 6.1 S60 5th Editionille porttaus

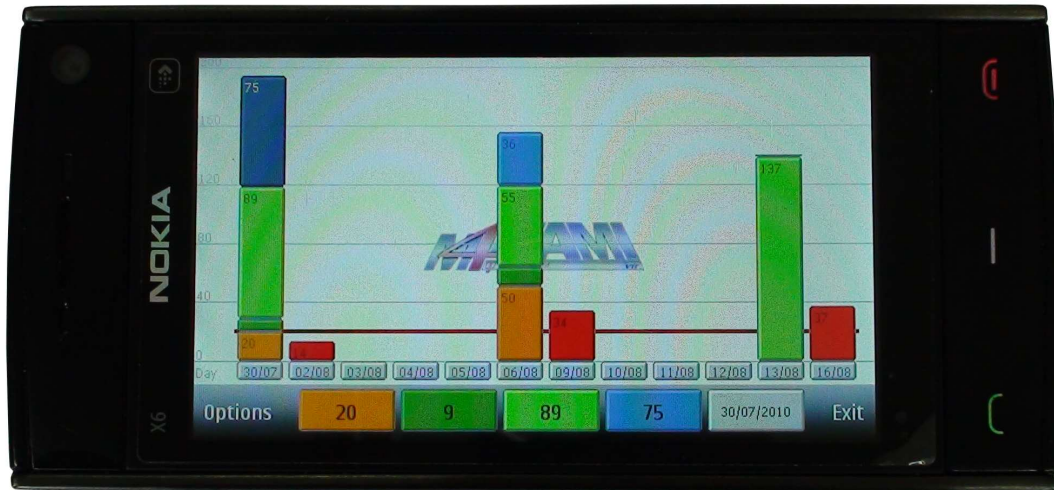
Nokia 9500 Communicator ja sitä hyödyntävä Series 80 -alusta ovat jo käyneet vanhaksi. Nokia on lopettanut Series 80 -alustan tukemisen. Viimeisille ilmestynyt laite on vuonna 2004 markkinoille tullut samainen Nokia 9500 Communicator. Nokialta on tosin tämän jälkeen ilmestynyt S60 3rd Editionia käyttävä Nokia E90 Communicator. Tätä mallia itse asiassa hyödynnettiin välivaiheena portattaessa sovellusta S60 5th Editionia tukevaan Nokia X6 -älypuhelimeen, joka on ominaisuuksiltaan jo huomattavasti tehokkaampi, kuin lähtökohtana käytössä ollut laite. E90 oli hyvä välivaiheen laite, sillä se vastasi ominaisuuksiltaan hyvin paljon sovelluksen alkuperäistä kohdelaitetta. Siksi sovelluksen näkymään tai dialogeihin ei tarvinnut tehdä juuri muutoksia porttauksen ensivaiheessa. Kuvassa 8 on esitetty Nokia 9500 Communicator, jolta sovellusta lähdettiin porttaamaan uuteen ympäristöön.



*KUVA 8. M4MAMI Nokia 9500 Communicatorissa*

Koska lähtökohtana käytetyssä laitteessa oli fyysinen näppäimistö, tuli sovelluksen porttauksen suunnittelussa ottaa huomioon fyysisen näppäimistön puuttumisen aiheuttamat vaatimukset. Näistä tärkein oli ennen kaikkea kyky pystyä valitsemaan näytöllä olevia elementtejä. Kuten kuvasta 8 nähdään, on osa lohkoista niin pieniä, että niitä olisi hyvin vaikea pystyä valitsemaan koskettamalla. Tästä kerrotaan lisää luvussa 6.2.5 (Kosketusnäyttötuen lisääminen).

Koska älypuhelimien näytön koko on hyvin rajallinen, on järkevää käyttää sovellusta siten, että se hyödyntää koko näytön. Tätä varten piti suunnitella erikseen alue menuvalikolle ja lisätä siihen halutut toiminnallisuudet. Tässä päädyin lopulta perinteiseen Symbianissa käytettyyn ratkaisuun, jossa menuvalikosta löytyvät sekä Options- että Exit-valinnat. Itse toteutetun menuvalikon ansiosta siitä pystyttiin luomaan sellainen, että se sopi hyvin yhteen sovelluksen päänäkymän kanssa. Tällä tavoin luodusta valikosta oli myös se etu, että siihen voitiin lisätä objekteja haluttu määrä, ilman että se pilasi sovelluksen ulkonäköä. Kuvassa 9 on esitetty M4MAMI portattuna Nokia X6 -älypuhelimelle.



KUVA 9. M4MAMI Nokia X6 -älypuhelimessa

Menuvalikon suunnittelussa käytiin läpi useita iteraatioita, joissa testattiin eri tapoja, joilla sovelluksen käytettävyyttä voitaisiin parantaa. Lopulta päädyttiin yllä olevaan ratkaisuun, joka oli näistä toimivin. Siinä menuvalikossa esitetään ne lohkot siltä päivältä, jolta on valittu joko päivä- tai tilauslohko. Esimerkiksi kuvassa 9 on valittuna vasemmassa laidassa oleva sininen lohko, ja kaikki kyseisen päivän lohkot on esitetty menuvalikossa. Lohko voidaan avata joko kaksoisnäpäyttämällä näytöllä olevaa lohkoa tai painamalla kerran menuvalikossa olevaa lohkoa.

Sovellukseen lisättiin myös ominaisuus, jonka avulla pystyttiin tutkimaan, liittyykö tilauksiin materiaali- tai kapasiteettipuutteita. Tämän toteutukseen vaadittiin lisäyksiä moneen paikkaan sovelluksessa. Ensinnä tuli ottaa huomioon uudet vaadittavat tietueet, joita tarvittiin MAMI-serveriltä haetun puutiedon tallentamiseen. Näitä tietoja varten täytyi luoda taulukko, johon tiedot talletettiin sen jälkeen, kun ne oli vastaanotettu GCM-viesteinä ja parseroitu. Serverille täytyi myös tehdä muutoksia, jotta uusi kysely voitiin palauttaa. Jotta tiedot voitiin esittää niiden haun jälkeen graafisesti, täytyi toteuttaa myös uusi dialogi-luokka sekä periyttää CCoeControl-luokka, jossa voitiin esittää puuterivit. Myös ne täytyi toteuttaa CCoeControl-luokasta periytettynä olioina. Näin niille voitiin toteuttaa kustomoitu piirto.



Tässä luvussa on esitetty ne vaiheet, jotka täytyi suorittaa sovelluksen porttaamisessa Series 80 -alustalta S60 5th Editionille. Kaikkea toteutuksessa lisättyä koodia ei kuitenkaan ole tarkoitus esittää tässä. Tarkoituksena on keskittyä siihen, mitä vaaditaan sovelluksen saamiseksi uudelle alustalle ja mitkä ovat olennaisia toimenpiteitä kosketusnäyttötuen aikaansaamiseksi. Sovellus piti lisäksi sisällään useita itserakennettuja dialogeja, joihin täytyi myös lisätä toteutus kosketusnäyttötuelle, päänäköymän lisäksi.

## **6.2 Vaiheet**

Sovelluksen porttaaminen aloitettiin hakemalla työkalut Nokia Forumista, ja kehitysympäristö pystytettiin luvussa 5 kuvatulla tavalla. Sovelluksen haluttiin tukevan edelleen myös näppäinpuhelimia, joten tästä syystä ladattiin myös S60 3rd Edition SDK.

Sovellus portattiin ensin S60 3rd Editionille, sillä muutokset S60 5th Editionia varten koskivat ainoastaan kosketusnäyttöominaisuuksien lisäämistä. Esitän tässä luvussa myös itse kosketusnäyttöominaisuuksien lisäämisen kannalta merkittävät kohdat, joita vaaditaan sovellukseen toteutettaviksi, sillä ne mahdollistavat sovelluksen käytön puhtaasti kosketusnäyttöllisissä puhelimissa.

## **6.3 Muutokset tiedostoihin**

Muutokset joita sovellukseen täytyi tehdä koskivat eräitä luokkia, joita ei ollut käytössä S60 3rd ja 5th Edition -versioissa. Muutoksia tuli tehdä resurssitiedostoihin, sovelluksen tyyppi tuli muuttaa ja sovellus täytyi myös rekisteröidä. Lisäksi tuli ottaa huomioon turvatoimien tuomat vaatimukset. Seuraavassa on esitetty ne tiedostot joihin täytyi tehdä muutoksia tai jotka täytyi lisätä projektiin:

- mmp-tiedostot
- sovelluksen aloituskohta

- rss-tiedosto
- <application\_name>\_reg.rss, sovelluksen rekisteröintitiedosto
- pkg-tiedosto sovelluksen .sis-paketin tekoa varten
- kosketusnäyttöominaisuuksien lisäämien

### 6.3.1 mmp-tiedostot

Projektin määrittelytiedostoon, eli mmp-tiedostoon, tuli muuttaa sovelluksen tyyppi. S80-versiossa sovelluksen tyyppinä käytettiin päätettä .app, kun se on ollut Symbian OS v9.0:sta lähtien GUI-sovelluksille (Graphical User Interface) .exe. Tästä seurasi seuraavat muutokset mmp-tiedostoon.

```
TARGET      M4MAMI.exe
TARGETTYPE  exe
```

Seuraava muutos koski UID3-arvoa, joka tuli muuttaa välille 0xA0000000-0xAFFFFFFF itse allekirjoitetuille sovelluksille. Ilman tätä muutosta sovellusta ei voida asentaa puhelimelle testattavaksi. Toinen vaihtoehto olisi ollut käyttää väliä 0xE0000000-0xEFFFFFFF, joka on tarkoitettu varta vasten sovellusten testaamiseen.

```
UID          0x100039CE 0xA0000001
```

Aiempi näistä UID-arvoista on UID2, joka on aina kyseinen GUI-sovelluksille. Tämän muutoksen jälkeen mmp-tiedostoon lisättiin vendorid sekä capability. Vendorid:tä tarvitaan, mikäli täytyy tehdä turvatarkistuksia sovelluksen exe-tiedostoa varten. Capabilityllä puolestaan tarkoitetaan jotain kykyä, jonka sovellus vaatii pystyäkseen suorittamaan tietyn toimenpiteen. Koska sovelluksen täytyi pystyä lukemaan tiedostosta sekä kirjoittamaan tiedostoon, täytyi lisätä ReadUserData- ja WriteUserData-capabilityt. Myös NetworkServices-capability täytyi lisätä, sillä sovelluksen tuli kyetä ottamaan langattomasti yh-

teys palvelimeen, josta ladata tarvittava informaatio näytettäväksi sovelluksessa, sekä tallentaa se puhelimen omaan tietokantaan.

```
CAPABILITY NetworkServices ReadUserData WriteUserData
```

Nämä ominaisuudet tuli lisätä jokaiseen sovelluksen käyttämään projektin määrittelytiedostoon. Resurssitiedostolle tuli asettaa uusi sijainti.

```
SOURCEPATH        ..\res\rss
START RESOURCE     M4MAMi.rss
HEADER
TARGETPATH        resource\apps
END
```

Sovelluksen rekisteröintitiedostoa varten tuli käyttää esikäsittelijäkomentoja, sillä tiedostolle käytetään hieman eri sijaintia sen mukaan, käännetäänkö sovellus emulaattorille vai kohdelaitteelle.

```
START RESOURCE     M4MAMI_reg.rss
#ifdef WINSCW
TARGETPATH        \private\10003a3f\apps
#else
TARGETPATH        \private\10003a3f\import\apps
#endif
END
```

Sovelluksen käyttämien bittikarttojen polut tuli asettaa uudestaan. Aiemmin polkuna oli ollut "\system\apps\AppName", joka muutettiin poluksi "\resource\apps"

```
START BITMAP       white.mbm
TARGETPATH        \resource\apps
HEADER
SOURCEPATH        ..\res\bitmaps\backgrounds
SOURCE            c16 white.bmp
END
```

Lisäksi kosketusnäyttöisiä puhelimia varten lisättiin makro, jonka vaikutuksesta tiettyjä osia sovelluksen lähdekoodista ehdollistetaan joko otettavaksi mukaan tai jätettäväksi pois itse sovelluksen binääristä. Tämän makron sisään on asetettu lähinnä sellaisia osia koodista, jotka liittyvät kosketustapah- tumien kiinniottamiseen sekä piirtoon, jossa on hieman eroavaisuuksia ei- kosketusnäyttölliseen versioon nähden.

```
MACRO __TUI_SUPPORTED__
```

Alla on esimerkki makron käytöstä sovelluksen lähdekoodissa. Metodin esitely sekä sen toteutus tulee ehdollistaa esikäsitteilykomentojen avulla. Eh- dollistamisessa käytetään seuraavanlaista syntaksia.

```
#ifdef __TUI_SUPPORTED__
void ClassName::DoSomething()
{
    // Suoritettava koodi...
}
#endif // __TUI_SUPPORTED__
```

### 6.3.2 Lähdekooditiedostot

Sovelluksen tyyppi muuttui Symbian OS v9.0:ssa exe-päätteiseksi. Samalla muuttui myös sovelluksen käynnistystapa. Aiemmin sovelluksen aloituskoh- tana oli E32Dll-metodi, kun se nykyisin on E32Main. Tämän seurauksena lisättiin E32Main-metodi sovellukseen. Mikäli yhteensopivuus vanhan S80- laitteen kanssa haluttaisiin säilyttää, voitaisiin ehdollistaa sovelluksen aloi- tuskohta. Tässä on kuitenkin esitetty itsessään vain aloituskohdan lisäys so- vellukseen sekä sen vaatima otsikkotiedoston lisäys.

```
#include <eikstart.h>
GLDEF_C TInt E32Main()
{
    return EikStart::RunApplication(NewApplication);
}
```

Sovelluksen UID3-arvon muutoksen seurauksena tuli tämä muutos lisätä myös lähdekooditiedostoon. CAknApplication-luokka sisältää AppDllUid()-metodin, jonka tulee palauttaa kyseinen sovelluksen UID. Sovelluksen Application-luokka on periytetty CAknApplication-luokasta.

```
const TUid KUidM4MAMI = { 0xA0000001 };
```

### 6.3.3 Resurssitiedostot

Sovellukset täytyy rekisteröidä, jotta ne tulevat näkyviin laitteen sovellusmenuvalikkoon. Rekisteröintiä varten täytyi luoda kokonaan uusi tiedosto, joka tyypillisesti nimetään seuraavalla tavalla: <application\_name>\_reg.rss. Tämä tiedosto sisältää sovelluksen lokalisoimattoman tiedon, joka on riippumaton siitä, mitä kieliversiota sovellus käyttää. Lokalisoinnin avulla sovellus voidaan asettaa tukemaan useita eri kieliä. \_reg.rss-tiedosto liitetään sovelluksen paketointi-tiedostoon (pkg-tiedosto). Sovelluksen rekisteröintiä varten luotu resurssi-tiedosto näyttää seuraavalta. (10.)

```
UID2 KUidAppRegistrationResourceFile
UID3 0xA0000001 // application UID Phone unprotected range
RESOURCE APP_REGISTRATION_INFO
{
    app_file = "M4MAMI";
    localisable_resource_file = "\\resource\\apps\\M4MAMI";
    localisable_resource_id = R_LOCALISABLE_APP_INFO;
}
```

Kyseisen tiedoston lisäksi tuli tarjota lokalisoitu informaatio sovellusta varten. Tällä tarkoitetaan kuvatekstejä sekä ikoneja. Lokalisoitu informaatio voidaan lisätä resurssitiedostoon alla olevalla tavalla.

```
RESOURCE LOCALISABLE_APP_INFO r_localisable_app_info
{
    short_caption = loc_title_short;
```

```

caption_and_icon =
{
    CAPTION_AND_ICON_INFO
    {
        caption = loc_title;
        number_of_icons = 2;
        icon_file="\\resource\\apps\\m4mami_reg.mif";
    }
};
}

```

Aiemmin sovelluksen ikonien luontiin voitiin käyttää AIF-resurssitiedostoa, joka sisälsi myös tiedon sovelluksen rekisteröinnistä. Tämä kuitenkin muuttui S60 3rd Editionin myötä. Nykyään ikoneiden luontiin voidaan käyttää esimerkiksi SDK:n mukana tulevaa mifconv-työkalua, jonka avulla voidaan tehdä sovelluksen ikoni menuvalikossa näytettäväksi. Työkalun käyttö on suhteellisen yksinkertaista, kun halutaan muuttaa tavallinen bittikartta-tiedosto (bmp-tiedosto) puhelimen tukemaan muotoon (mif-tiedosto). Mif-tiedoston luonti onnistuu seuraavalla tavalla.

```

c:<miftool path>mifconv.exe c:<targetpath>my.mif /c24,1
c:<bitmap path>hke.bmp

```

Tämä luo my.mif- sekä my.mbm-tiedostot c:<targetpath>-hakemistoon. Sieltä ne tulee hakea sovelluksen paketoititiedostoon.

Lisäksi muutoksia täytyi tehdä tiettyihin sovelluksen käyttämiin dialogeihin, jotka oli suunniteltu varta vasten Nokia 9500 Communicatorille. Tässä kommunikaattorissa on kaikkiaan neljä fyysistä nappia, joilla voitiin ohjata menuvalintoja, kun S60 puolestaan tukee ainoastaan kahta niin sanottua softkeytä. Niinpä tiettyihin dialogeihin täytyi lisätä menuvalikot kompensoimaan puuttuvia nappeja.

### 6.3.4 Sovelluksen paketoitiedosto

Jotta sovellus voidaan asentaa puhelimelle, täytyy luoda sovelluksen asennuspaketti (sis-tiedosto). Tämä puolestaan luodaan käyttäen sovelluksen paketoitiedostoa. Pkg-tiedosto sisältää muun muassa tiedon sovelluksen tukemista kielistä, sovelluksen omistajan sekä tuetut alustat. Pkg-tiedostoon asetetaan kaikki sis-paketin vaatimat tiedot, kuten sovelluksen exe-tiedosto, rsc-tiedosto sekä reg\_rsc-tiedosto. Koska työstämäni sovellus käyttää myös ikoneja, tuli nämä tiedostot lisätä pkg-tiedostoon alla esitetyllä tavalla.

```
"C:<sourcepath>my.mif" -"!:\resource\apps\m4mami_reg.mif"  
"C:<sourcepath>my.mbm" -"!:\resource\apps\m4mami_reg.mbm"
```

Tässä `sourcepath` on se polku, johon `mifconv`-työkalun avulla luotiin kyseiset tiedostot. Jälkimmäinen osa

```
- "!:\resource\apps\m4mami_reg.mif" puolestaan kertoo sijainnin
```

jonne tiedostot asetetaan puhelimessa. Koska sovellukseen rakennettiin tuki myös kosketusnäytölle, lisättiin pkg-tiedostoon myös rivi, joka kertoo, että myös S60 5th Editionin sisältävät laitteet ovat tuettuja. Kyseisen alustan UID on `0x1028315F`.

```
;Supports S60 release 5.0  
[0x1028315F], 0, 0, 0, {"Series60ProductID"}
```

### 6.3.5 Kosketusnäyttötuen lisääminen

Sovelluksen haluttiin tukevan nykyisiä kosketusnäytöllisiä puhelimia. Siksi tuli ottaa huomioon muutamia asioita, kuten elementtien koot, joita tuli kyetä valitsemaan sovelluksessa, liikkuminen näytöllä, kun fyysisiä näppäimiä ei ole tarjolla, sekä piirron reagoiminen laitteen asentoon (skaalautuva UI) eli siihen, onko laite pystyasennossa vaiko vaaka-asennossa.

Laitteen asennon muutokset saadaan kiinni `HandleResourceChange()`-metodissa, joka tuli toteuttaa näkymä- eli `View`-luokkaan. Metodissa otetaan kiinni se tapahtuma, kun asettelua muutetaan. Tällöin muutetaan kontrollin

koko halutuksi. Kun kontrollin kokoa on muutettu, kutsutaan View-luokan `SizeChanged()`-metodia. Mikäli näkymä sisältää kontrolleja, tulee kontrollien sijainti asettaa kyseisessä metodissa. Alla on esimerkki `HandleResourceChange()`-metodin toteutuksesta.

```
void CM4MamiAppView::HandleResourceChange(TInt aType)
{
    //call base class implementation
    CCoeControl::HandleResourceChange(aType);

    if ( aType==KEikDynamicLayoutVariantSwitch )
        {
            SetExtentToWholeScreen();
        }
}
```

Toinen oleellinen asia, joka pitää ottaa huomioon, on kosketustapahtumien kiinniottaminen. Kaikki koodi, joka liittyy suoraan kosketustapahtumien kiinniottamiseen tai niistä aiheutuviin tapahtumiin, sisällytettiin mmp-tiedostossa määritellyn `__TUI_SUPPORTED__`-makron sisään. Kosketustapahtumat otetaan kiinni View-luokan `HandlePointerEventL()`-metodissa. Metodi saa parametrina `TPointerEvent`-tietueen, joka sisältää muun muassa tiedon tapahtuman tyypistä sekä sijainnista. Näiden tietojen avulla pystyttiin toteuttamaan kosketusnäyttöominaisuuksien tukeminen. Alla on karsittu esimerkki `HandlePointerEventL()`-metodista.

```
void CM4MamiAppView::HandlePointerEventL(const
    TPointerEvent& aPointerEvent)
{
    TPoint pos = aPointerEvent.iPosition;
    TInt optionsRectHeight = iDrawer->OptionsRect().Height();
    Tpoint startPoint(0,iDrawRect.iBr.iY-optionsRectHeight);
    TPoint endPoint(iDrawRect.iBr);
    TRect menuRect(startPoint, endPoint);

    // If button is pressed elsewhere than on menu rectangle
```



```

// -> start listening for long taps.

if(!menuRect.Contains(pos))
    iLongTapDetector->PointerEventL(aPointerEvent);

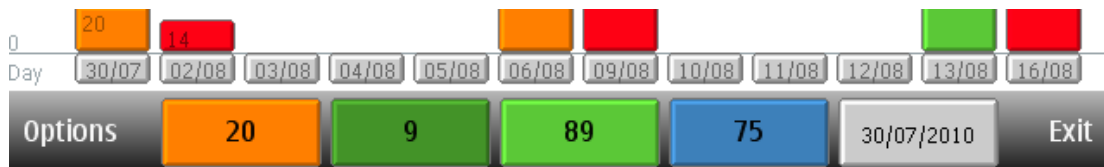
if(aPointerEvent.iType == TPointerEvent::EButton1Down)
    {
    if(iReceiveKeyEvents)
        {
        iDragPoints.Reset();
        SelectTimeOrBarBlock(pos);
        }
    }
else if(aPointerEvent.iType == TPointerEvent::EDrag)
    {
    if(iIsReady)
        {
        if(!iLongTapDetected){
            iDragPoints.Append(pos);
        }
        else{
        /*
        * If long tap is detected we could move the
        * barBlocks in this else statement.
        */
        }
        }
    }
/*
* Rest of the function is missing from this example.
*/
}

```

Selvitys edellisestä metodista: Otetaan parametrina vastaan TPointerEvent, joka sisältää tiedon sijainnista, jossa näyttöä kosketettiin. Mikäli tämä

sijainti ei ole Options-menun päällä hyväksytään myös LongTap-tapahtuma. Tämä tapahtuma syntyy, kun sormea pidetään näytöllä tietty aika paikallaan. Kuten aiemmin totesin, TPointerEvent sisältää myös tapahtuman tyyppin, josta selviää muun muassa painettiin juuri näyttöä, vapautettiin sormi näytöltä vai raahattiin sormea näytöllä. Näiden perusteella päätetään, mitä sovelluksen tulee seuraavaksi tehdä.

Kun alkuperäistä sovellusta aikanaan kehitettiin, ei ollut tarvetta ottaa huomioon tiettyjen näytölle piirrettävien elementtien kokoa, joka on yksi merkittävä asia kosketusnäytöllistä sovellusta suunniteltaessa. Elementtien on oltava riittävän isoja, jotta niitä pystyttäisiin valitsemaan sormella käyttäen. Tästä johtuen jouduin kehittämään ratkaisun, jonka avulla voitiin valita hyvin pieniä elementtejä näytöltä ilman ongelmia. Kuten kappaleessa 2 todettiin, M4MAMi esittää tietyllä statuksella olevia tilauksia eri väreillä. Mikäli näytölle piirretään hyvin pieniä lohkoja useita päällekkäin, on niitä hyvin vaikea saada avattua. Siksi lisäsin näytön alalaitaan päivän tai viikon lohkot riittävän isoina valittaviksi. Lohkot on esitetty kuvassa 10.



*KUVA 10. M4MAMIn kosketusnäyttöversiossa käytetyt apulohkot*

## 7 JATKOKEHITYS

M4MAMI jatkokehityskohteet liittyvät paitsi itse sovelluksen jatkokehittämiseen myös Qt-tuen rakentamiseen. Qt on graafinen käyttöliittymäkirjasto, joka tulee olemaan ainoa työkalu Symbian-sovelluskehityksessä. Tämä tarkoittaa sitä, että perinteinen S60:n käyttämä AVKON tulee joka tapauksessa väistymään Qt:n tieltä. Tämän vuoksi olisi tärkeää ottaa myös Qt mukaan M4MAMIn kehitykseen.

Qt on alkujaan norjalaisen Trolltechin kehittämä tuote alustasta riippumattomien graafisten käyttöliittymien ja ohjelmistojen luontiin. Qt:n se julkaisi vuonna 1996. Sitä on käytetty muun muassa seuraavissa projekteissa: Google Earth, KDE, Opera ja Skype. Qt-kirjastoon on sisäänrakennettu tuki C++-ohjelmointikielelle, mutta sitä voidaan ohjelmoida myös useilla muilla ohjelmointikielillä. Näitä ovat muun muassa Java, Python, Ruby sekä PHP. Javascript-tuki tulee löytymään Qt 4.7-versiosta samoin kuin Nokian kehittämä QML-kuvauskieli, jota käytetään kuvaamaan sovelluksen käyttöliittymää (3, hakusana Qt (framework); 11.)

Nykyisellään Nokian käyttämää Symbiania on syytetty siitä, että sovellusten ohjelmointi sille on vaikeaa, ja niiden porttaaminen useille eri puhelinmalleille on ollut työlästä. Nokia osti Trolltechin vuonna 2008 vastatakseen tähän ongelmaan hankkimalla työkalun, jolla voi helposti kehittää useissa eri laitteissa suoraan toimivia sovelluksia. Näin se toivoo houkuttelevansa puolelleen kehittäjiä, jotka ovat kiistatta tärkeitä myös itse laitteiden menestymisen kannalta nykypäivänä.

Se, että Qt tulee korvaamaan AVKONin, ei tarkoita sitä, että kaikki aikaisemmin toiminut natiivi Symbian-koodi tulisi kirjoittaa uusiksi. Qt:tä voidaan ajaa yhdessä myös natiivin Symbian C++:n rinnalla. Ainoastaan käyttöliittymä joudutaan kirjoittamaan uusiksi. Sovellusten toimintalogiikan sisältävät engine-luokat voivat pysyä sellaisenaan, sillä ne eivät sisällä mitään AVKONiin liittyvää.

## 7.1 Qt:n alustariippumattomuus

Kuten aiemmin mainitsin, on Qt alustariippumaton ja mahdollistaa sovellusten kehittämisen useille eri alustoille. Tämä mahdollistaisi M4MAMIn käytön monipuolisemmalla laitekannalla. Symbianin lisäksi sovellusta voitaisiin ajaa myös tulevilla Meego-laitteilla. Meego on Nokian ja Intelin yhteistyöprojekti, jonka tarkoituksena on kehittää Linux-pohjainen avoin käyttöjärjestelmä mobiililaitteille. Meegoa on tarkoitus pystyä käyttämään useilla eri alustoilla joita ovat muun muassa älypuhelimet ja tablet-laitteet. Ensimmäisiä laitteita voidaan odottaa vuoden 2011 aikana. (13.)

Tablet-laitteissa on nykyään noin 10 tuuman kokoinen kosketusnäyttö, joka on yli kaksinkertaisesti sen mitä suurimmat älypuhelimet tarjoavat. Tästä johtuen M4MAMIn graafista ulkoasua voitaisiin toteuttaa vastaamaan enemmän työpyötäversiota sekä samalla lisätä sen esittämää informaatiota. Tosin tässä täytyisi ottaa huomioon lähtökohta, jossa M4MAMIn engine on toteutettu natiivia Symbian C++ -kieltä käyttäen, jota Meego ei tue. Sen seurauksena porttaus aiheuttaisi isomman työn verrattuna Symbian porttaukseseen, jossa ainoastaan graafinen käyttöliittymä tulisi toteuttaa uusiksi.

## 7.2 Qt:n hyödyntäminen

Qt:hen on rakennettu tehokas Graphics View -arkkitehtuuri, jolla avulla voidaan piirtää kaksiulotteisia graafisia objekteja piirtopinnalle. Piirtopintaa voidaan lähentää ja loitontaa sekä kiertää. Graphics View -arkkitehtuuri koostuu kolmesta eri pääluokasta, jotka ovat QGraphicsView, QGraphicsScene ja QGraphicsItem sekä siitä periytyneet luokat. Näitä kaikkia luokkia voidaan käyttää sellaisenaan. Mikäli sovellukselle halutaan räätälöidä ulkoasu, tulee joko osa tai kaikki niistä periyttää riippuen sovellukselle asetetuista vaatimuksista. QGraphicsView-luokka visualisoi QGraphicsScene-luokan sisällön, joka koostuu joukosta QGraphicsItem-luokan olioita. QGraphicsScene tarjoaa helppokäyttöisen rajapinnan QGraphicsItemien hallintaan. Se huolehtii niiden tilasta, kuten valinnan ja fokuksen käsittelystä.

Monet ominaisuudet toimivat valmiiksi sisäänrakennettuina tässä arkkitehtuurissa, esimerkiksi QGraphicsItemien valinta, jota sovelluksen kehittäjän ei tarvitse erikseen koodata. Mikäli hän haluaa toteuttaa oliolle kustomoidun piirron, tutkii hän olion piirtometodissa ainoastaan sen parametrin perusteella onko olio valittuna vai ei. Periyttämisessä tulee ylikirjoittaa myös olion geometriaa koskevat metodit, jotta arkkitehtuuri osaa suorittaa sen valinnan oikein. Itse asiassa boundingRect-metodi on pakollista periyttää sillä se on puhdas virtuaalinen metodi, joka tarkoittaa sitä, että ilman sen toteuttamista ei luokasta voitaisi instantoida olioita. QGraphicsItemille voidaan asettaa erilaisia ominaisuuksia sen tarpeiden mukaan. Näitä ovat muun muassa seuraavat: valinta, liikuttaminen, kaksoisnäpätys, raahaus ja pudotus, ryhmitteily ja törmäyksen tunnistus. Monesta näistä ei välttämättä olisi hyötyä M4MAMIn kehityksen kannalta, mutta mahdollisuus raahata voisi olla yksi hyvä hyödyllinen ominaisuus. Tämä mahdollistaisi tilausten siirtämisen päivältä toiselle. Toki sitä varten tulisi lisätä ominaisuus, joka mahdollistaisi valinnan mitä tilauksia siirretään, jotta kaikkia samalla statuksella olevia tilauksia ei siirrettäisi yhdessä. (12.)

Graphics View -arkkitehtuuria voitaisiin käyttää paitsi sovelluksen varsinaisen päänäkymän piirrossa, niin myös niiden dialogien piirrossa, jotka esittävät tilauksia. Sitä voitaisiin hyödyntää muun muassa käyttämällä QGraphicsItemien valintaominaisuutta, joka mahdollistaisi tiettyjen tilausrivien valitsemisen siirtoa varten itse sovelluksen päänäkymässä.

## 8 POHDINTA

Opinnäytetyössä toteutettiin tuotannosuunnittelu- ja kapasiteetilaskenta-sovelluksen porttaus Nokia 9500 Communicatorin käyttämältä S80-alustalta nykyisille Nokian kosketusnäyttöpuhelimille, jotka käyttävät S60 5th Editionia käyttöliittymänään. Aihe tähän saatiin työn tilaajalta, Qprojects Oy:ltä, joka halusi M4MAMIn tukevan uusinta laitekantaa. Opinnäytetyön tekohetkellä Symbian ^3 -versioon pohjautuvia laitteita ei ollut vielä markkinoilla, joten porttauksen kohteeksi valittiin S60 5th Edition. Nyt kun opinnäytetyö alkaa olemaan hyväksymistä vaille valmis, on Symbian ^3 -versioon porttaaminen myös suoritettu. Alustavasti Qt on suunniteltu otettavaksi mukaan jatkokehittelyä varten.

Kun aloin tehdä opinnäytetyötä, ei minulla ollut vielä lainkaan kokemusta porttaamisesta, saati siitä, minkälaisia eroja käytettävien alustojen välillä oli ja mitä muutoksia sovellus vaatisi toimiakseen uudessa laitteessa. Tämän lisäksi edellisestä kerrasta, kun olin ollut Symbianin parissa tekemisissä, oli kulunut jo lähes kaksi vuotta. Jouduin siis kertaamaan myös Symbian-ohjelmointia koskevia asioita. Ennen varsinaisen porttaamisen aloittamista tutustuin siihen dokumentaatioon, mitä M4MAMlsta oli. Niitä olivat muun muassa sovelluksen toimintaa kuvaavat dokumentit, luokkakaavio sekä viestiyhteyskaaviot. Luokkakaavion perusteella sain jo hyvän käsityksen siitä, millä tavalla sovelluksen luokat ovat toistensa kanssa sidoksissa.

Varsinainen porttaaminen olikin alussa kääntäjävirheiden syiden etsimistä ja niiden korjaamista. Muutokset eivät välttämättä olleet järin suuria, mutta ennen kuin ratkaisut näihin löytyivät, täytyi nettiä käyttää ahkerasti. Alkuvaiheessa Google oli kultaakin kalliimpi apuväline virheiden syiden paikantamisessa. Dialogit ja eräät niissä käytetyt kontrollit ja niiden toiminta aiheuttivat myös päänvaivaa, sillä jotkin käytetyt kontrollit vaativat koodin lisäämistä niiden tunnistamista varten dialogissa. Ilman tätä sovellus kaatui dialogia avattaessa. Sovelluksen asentamiseksi puhelimeen oli myös tullut pieniä muu-

toksia. Sovellus tuli ensinnäkin rekisteröidä, jotta sen sai näkyvään sovellusvalikossa. Myös sovelluksen ikoni täytyi luoda uudella tavalla. Näihin ongelmiin löytyi kuitenkin hyvin apua Forum Nokia -sivuilta.

Sen jälkeen, kun sain sovelluksen portattua S60 3rd Editionille, aloitin S60 5th Editionille porttaamisen ja varsinaisen kosketusnäyttötuen lisäämisen. Kosketusnäyttötuellista versiota varten tuli myös kehittää sovellukseen lisäominaisuus lohkojen valitsemista varten, jotta sen käyttö olisi sujuvaa. Itse kosketusnäyttötuen lisääminen ei ollut kovinkaan hankalaa enää tässä vaiheessa, sillä sovellus oli ehtinyt tulla tässä vaiheessa jo toimintansa puolesta tutuksi. Tässä vaiheessa sovellukseen alettiin myös toteuttaa uusia ominaisuuksia, joihin kuului muun muassa puutetietojen esittäminen. Ennen uusien ominaisuuksien lisäämistä tein aina suunnitelman vaadittavia lisäyksiä varten toteutetusta ohjelmakoodista sekä sen suorittamasta logiikasta. Tämä vähensi huomattavasti virheiden syntymistä, sillä mahdolliset ongelmat tulivat esille jo itse suunnitteluvaiheessa.

Sovellusta kehitettiin siten, että aina kun uusia ominaisuuksia lisättiin, testattiin niiden toiminta sekä S60 3rd että 5th Editionilla. Joissain tapauksissa täytyi tehdä erilaiset ratkaisut eri versioihin, mutta nämä koskivat lähinnä eroja piirroissa. Projektin loppuvaihe oli lähinnä sovelluksen testausta ja virheiden korjaamista. Esimerkiksi sovelluksen offline-tilan toiminnasta löytyi useita ongelmia, jotka saattoivat aiheuttaa sovelluksen kaatumisen tai tietojen esittämisen väärin. Offline-tilassa paransin muun muassa tietokannan käsittelyä päänäkyvän lohkojen hauissa. Joissain tapauksissa virheiden korjaaminen oli haasteellista, sillä jonkin vian korjaaminen S60 5th Edition -versiossa aiheutti toimimattomuuden S60 3rd Edition -versiossa.

Työ oli kaiken kaikkiaan mielenkiintoinen ja sellainen, jota etsin, sillä halusin opinnäytetyön aiheen olevan jollain tapaa sidoksissa mobiiliohjelmointiin. Qt-ohjelmointi olisi ollut mielenkiintoinen lisä, mutta itse työ olisi paisunut liian laajaksi, mikäli sen olisi ottanut mukaan. Työn tuloksena saatiin toimiva kosketusnäyttötuellinen versio M4MAMlsta.

## LÄHTEET

1. Samsung terminates support for Symbian. 2010. Saatavissa:  
<http://www.engadget.com/2010/10/01/samsung-terminates-support-for-symbian-development/>. Hakupäivä 1.10.2010.
2. Sony Ericsson hylkää Symbian-puhelimet. 2010. Saatavissa:  
[http://www.tietokone.fi/uutiset/sony\\_ericsson\\_hylkaa\\_symbian\\_puhelimet](http://www.tietokone.fi/uutiset/sony_ericsson_hylkaa_symbian_puhelimet).  
Hakupäivä 24.9.2010.
3. Wikipedia. 2010. The Free Encyclopedia. Saatavissa:  
[http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page). Hakupäivä 15.9.2010.
4. History of Symbian. 2010. The Symbian Foundation Community Home.  
Saatavissa: <http://www.symbian.org/about-us/history-symbian>. Hakupäivä  
15.9.2010.
5. Power Management. 2009. Saatavissa:  
[http://developer.symbian.org/wiki/index.php/Symbian\\_OS\\_Internals/15.\\_P  
ower\\_Management](http://developer.symbian.org/wiki/index.php/Symbian_OS_Internals/15._Power_Management). Hakupäivä 17.9.2010.
6. Introducing EKA2. 2009. Saatavissa:  
[http://developer.symbian.org/wiki/index.php/Symbian\\_OS\\_Internals/1.\\_In  
troducing\\_EKA2](http://developer.symbian.org/wiki/index.php/Symbian_OS_Internals/1._Introducing_EKA2). Hakupäivä 17.9.2010.
7. Harrison, Richard – Shackman, Mark 2007. Symbian OS C++ for Mobile  
Phones. Application Development for Symbian OS v9. The Atrium,  
Southern Gate, Chichester, West Sussex PO19 8SQ, England: John  
Wiley & Sons, Ltd.
8. Gamma, Erich – Helm, Richard – Johnson, Ralph – Vlissides, John 2004.  
Design Patterns. Elements of Reusable Object-Oriented Software. 31.,



uudistettu pianos. Indianapolis, USA: Addison-Wesley, Pearson Education.

9. Moving to Windows Vista. 2010. Saatavissa:  
[http://wiki.forum.nokia.com/index.php/Moving\\_to\\_Windows\\_Vista](http://wiki.forum.nokia.com/index.php/Moving_to_Windows_Vista).  
Hakupäivä 15.7.2010.
10. C++ Developer's Library 1.4. 2008. Forum Nokia. Saatavissa:  
[http://www.forum.nokia.com/document/Cpp\\_Developers\\_Library/?content=GUID-FE9017F4-4197-472F-A3E7-267169A51ABD.html](http://www.forum.nokia.com/document/Cpp_Developers_Library/?content=GUID-FE9017F4-4197-472F-A3E7-267169A51ABD.html).  
Hakupäivä 25.7.2010
11. Introduction to the QML language. 2010. Saatavissa:  
<http://doc.qt.nokia.com/4.7-snapshot/qdeclarativeintroduction.html>.  
Hakupäivä 24.9.2010.
12. Graphics View Framework. 2010. Saatavissa:  
<http://doc.trolltech.com/4.7/graphicsview.html>. Hakupäivä 5.10.2010.
13. Ei Meego-kännyköitä tänä vuonna. 2010. Saatavissa:  
<http://www.itviikko.fi/uutiset/2010/10/06/intel-ei-meego-kannykoita-tana-vuonna/201013881/7>. Hakupäivä 6.10.2010.
14. Symbian 4 –puhelimia ei tulekaan. 2010. Saatavissa:  
<http://www.digitoday.fi/mobiili/2010/10/21/nokia-symbian-4--puhelimia-ei-tulekaan/201014651/66>. Hakupäivä 21.10.2010.
15. M4MAMI. 2010. Saatavissa:  
[http://www.qprojects.fi/ind.php?c=content\\_fin/Products/MAMI/m4mami](http://www.qprojects.fi/ind.php?c=content_fin/Products/MAMI/m4mami).  
Hakupäivä 15.9.2010.
16. Gcm protocol requirement specification. 2010. Saatavissa: Ei saatavissa.  
Hakupäivä 16.9.2010.

17. The Symbian^4 Overview. 2010. Saatavissa:  
<http://developer.symbian.org/wiki/index.php/Symbian^4>. Hakupäivä  
5.10.2010.
  
18. Desing Patterns in Symbian. 2009. Saatavissa:  
[http://wiki.forum.nokia.com/index.php/Design\\_Patterns\\_in\\_Symbian](http://wiki.forum.nokia.com/index.php/Design_Patterns_in_Symbian).  
Hakupäivä 17.9.2010
  
19. MAMI. 2010. Saatavissa:  
[http://www.qprojects.fi/ind.php?c=content\\_fin/Products/MAMI](http://www.qprojects.fi/ind.php?c=content_fin/Products/MAMI).  
Hakupäivä 15.9.2010.