

Sari Huotari

MULTIMEDIAVIESTINTÄ SYMBIAN OS ÄLYPUHELIMISSA

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Tradenomikoulutus
Syksy 2004



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Ala Luonnontieteet	Koulutusohjelma Tietojenkäsittely
Tekijä(t) Sari Huotari	
Työn nimi Multimediaviestintä Symbian OS älypuhelimissa	
Vaihtoehtoiset ammattiopinnot	Ohjaaja(t) Arto Karjalainen
Aika 30.9.2004	Sivumäärä 1 + 38
<p>Tämän opinnäytetyön tarkoituksena oli tutustua kolmannen sukupolven älypuhelimissa käytettävään Symbian OS käyttöjärjestelmään. Opinnäytetyö keskittyy tarkastelemaan järjestelmää sen toiminnallisuuden osalta yksityiskohtaisen teknisen tarkastelun sijaan. Lisäksi työssä tutustutaan Symbian OS käyttöjärjestelmän päällä toimiviin käyttöliittymämalleihin, joista tarkemman tarkastelun kohteena on Nokian lisenssoima Series 60 ohjelmistoalusta. Työn teoriaosa käsittelee lisäksi mobiilipäätelaitteiden ohjelmistotestaamista.</p> <p>Opinnäytetyössä on käytetty lähteenä pääosin internetistä saatavaa materiaalia, koska Symbianista on olemassa vielä melko vähän kirjallista aineistoa. Series 60 sovellusten ohjelmoinnin opiskelemiseen käytettiin julkaisematonta kurssimateriaalia.</p> <p>Opinnäytetyön käytännön osassa tarkoituksena oli ohjelmoida sovellus, jolla voidaan vastaanottaa ja käsitellä tekstiä, kuvaa tai ääntä sisältäviä multimediaviestejä. Lisäksi tarkoituksena oli testata sovellus ohjelmistotestaamisen näkökulmasta. Sovelluksen tuli toimia Symbian OS käyttöjärjestelmän ja Series 60 ohjelmistoalustan sisältävissä päätelaitteissa. Projekti toteutettiin Symbian C++ ohjelmointikielellä ja siihen lisättiin mukaan Forum Nokian internet sivustolta löytyneen MMSDemo1 -esimerkkisovelluksen sovellusmoottoriosa.</p> <p>Lopputuloksena syntynyt MMS sovellusta voidaan helposti jatkokehittää esimerkiksi toteuttamalla kaikenlaisten multimediaviestien käsittely ja vastaanotto. Myös sovellusarkkitehtuuri voidaan muuttaa toisenlaiseksi ja käyttöliittymäkomponentteja voidaan vaihdella.</p>	
Luottamuksellisuus	Julkinen
Hakusanat	Symbian, multimedia, ohjelmistotestaus
Säilytyspaikka	Kajaanin ammattikorkeakoulun kirjasto



**Kajaanin
ammattikorkeakoulu**

ABSTRACT OF THE FINAL YEAR PAPER

Faculty Natural Sciences	Degree programme Data Processing
Author(s) Sari Huotari	
Title Multimedia Messaging in Symbian OS Smart Phones	
Alternative professional studies	Instructor(s) Arto Karjalainen
Date 30.9.2004	Total number of pages 1 + 38
<p>The purpose of this final year project was to get acquainted with the Symbian operating system used in the third generation smart phones. The project focuses on its functionality instead of detailed technical description. In addition, user interface models working on top of the Symbian operating system are introduced and one of them, Nokia's Series 60 software development environment has been considered in more detail. Software testing of the mobile devices is the third part of this final year project.</p> <p>Sources of this final year project were mainly taken from the internet, because there are quite few books available of the Symbian OS. Unpublished course material was used for studying the Series 60 application programming.</p> <p>In the practice phase of the project the purpose was to create an application for receiving and handling multimedia messages consisting of text, images or audio and test it according to the software testing principles. The application was meant for mobile devices using the Symbian operating system and the Series 60 software platform. The application was implemented by the Symbian C++ programming language using the MMSDemo1 example program found from the Forum Nokia internet pages. Its application engine was utilized in this project.</p> <p>The final MMS application can be easily developed further, for example by implementing the receiving and handling of all kinds of multimedia messages. Also the application architecture and the user interface components can be changed.</p>	
Confidentiality status	Public
Keywords	Symbian, multimedia, software testing
Deposited at	Kajaani Polytechnic Library

SISÄLLYS

SYMBOLILUETTELO

1	JOHDANTO	1
2	SYMBIAN YHTIÖ	2
3	SYMBIAN OS ÄLYPUHELINTEN KÄYTTÖJÄRJESTELMÄNÄ	4
3.1	Vaatimukset älypuhelimien käyttöjärjestelmälle	4
3.2	Symbian OS:n kuvaus	5
3.2.1	Käyttöjärjestelmän rakenne	5
3.2.2	Symbian OS 7.0s version toiminnallisuus	6
4	SERIES SARJAN OHJELMISTOKEHITYSALUSTAT	8
4.1	Series 40	8
4.2	Series 60	9
4.2.1	Käyttöliittymä	9
4.2.2	Sovellusten perusrakenne	10
4.3	Series 90	11
5	MULTIMEDIAVIESTINTÄ	12
5.1	MMS viestien kulku tietoverkossa	12
5.2	MMS viestin rakenne	13
6	MOBIILI PÄÄTELAITTEEN OHJELMISTOTESTAAMINEN	15
6.1	Testaamisen tasot	15
6.1.1	Yksikkötestaus	16
6.1.2	Integroititestausta	17
6.1.3	Järjestelmä- ja hyväksymistestausta	17
6.2	Emulaattori	18

7	SOVELLUS MULTIMEDIAVIESTIEN VASTAANOTTAMISEEN	20
7.1	Tavoitteet	20
7.2	Suunnittelu	21
7.2.1	Luokkakaaviot	22
7.2.2	Sekvenssikaaviot	24
7.3	Toteutus	27
7.3.1	Lähtökohta	28
7.3.2	Omat lisäykset	28
7.3.3	Ongelmatilanteet	29
7.4	Testaus	30
7.5	Kehitysehdotukset	32
8	POHDINTA	33
	LÄHTEET	36

SYMBOLILUETTELO

API	Application Programming Interface: sovellusohjelmointirajapinta tai liityntä, vastaanottimessa toimivien sovellusten liittymä käyttöjärjestelmään
EPOC	Psionin kehittämä langattomien päätelaitteiden käyttöjärjestelmä
HTML	HyperText Markup Language: internetissä käytetty kuvauskieli sivustojen rakennuksessa
GPRS	General Packet Radio Service: tiedonsiirtotapa erilaisille langattomille sovelluksille, joissa tietoa siirretään ns. paketteina
MIDP	Mobile Information Device Profile: profiili, joka määrittelee rajapinnat esimerkiksi tiedon esittämistavoille matkapuhelimen sisällä
MIME	Multipurpose Internet Mail Extensions: standardi muun kuin tekstidatan siirtämiseen internetin yli
MMF	Multi-threaded Multimedia Framework: Symbian 7.0 versiossa toimiva multimediakirjasto
MMS	Multimedia Messaging Service: standardi tekstiä, grafiikkaa, kuvia, ääntä tai videoleikkeitä sisältävien viestien lähettämiseen ja vastaanottamiseen
MMSC	Multimedia Messaging Service Center: multimediaviestien viestikeskus
SDK	Software Development Kit: ohjelmistokehityspaketti, joka sisältää mm. erilaisia dokumentteja ja työkaluja sovellusten ohjelmointiin
SMIL	Synchronized Multimedia Integration Language: multimedian esitystä varten kehitetty HTML:n (Hypertext Markup Language) kaltainen ohjauskieli
SyncML	Protokolla tiedon synkronointiin erilaisten langattomien ja verkossa toimivien laitteiden välillä
URL	Universal Resource Locator: osoite, joka viittaa tiettyyn internetistä löytyvään informaatioon

WAP	Wireless Application Protocol: WAP Forumin määrittelemä standardi, joka mahdollistaa internetin käytön langattomissa päätelaitteissa
WCDMA	Wideband Code Division Multiple Access: CDMA:sta kehitetty kolmannen sukupolven matkapuhelinjärjestelmä
XHTML	Extensible Hypertext Markup Language: HTML kuvauskielen seuraaja
3GPP	The 3rd Generation Partnership Project: mm. operaattoreiden ja laitevalmistajien perustama maailmanlaajuinen tietoliikennealan yhteistyöelin, joka on määritellyt 3GPP-standardin multimediasisällön siirtämiseen nopeissa matkapuhelinverkoissa

1 JOHDANTO

Viime vuosina matkapuhelimet ovat kehittyneet huimaa vauhtia ja niihin on kehitetty monenlaisia uusia palveluja. Uudet palvelut vaativat uudenlaisen toimintaympäristön, minkä vuoksi kolmannen sukupolven matkapuhelimiin on täytynyt kehittää uudenlainen käyttöjärjestelmä. Uusien ominaisuuksien lisäksi käyttöjärjestelmän rakenne on kehitetty avoimen standardin mukaisesti. Tämä luo valtavasti uusia mahdollisuuksia, koska jokainen ohjelmointitaitoinen voi halutessaan kehittää omia sovelluksiaan kolmannen sukupolven älypuhelimiin.

Matkapuhelinpalveluista tekstiviestit ovat saavuttaneet suuren suosion. Palvelu muuttuu multimediaviestiksi, kun viestiin halutaan liittää grafiikkaa, kuvia, ääntä tai videoleikkeitä. Multimediapalvelut ovat uusinta uutta eivätkä ne ole vielä kovin yleisiä. Uusimmat matkapuhelinmallit sisältävät jo digitaalisen kameran, jolla voi ottaa kuvia ja lähettää niitä multimediaviestinä toiseen matkapuhelimeen tai sähköpostiin.

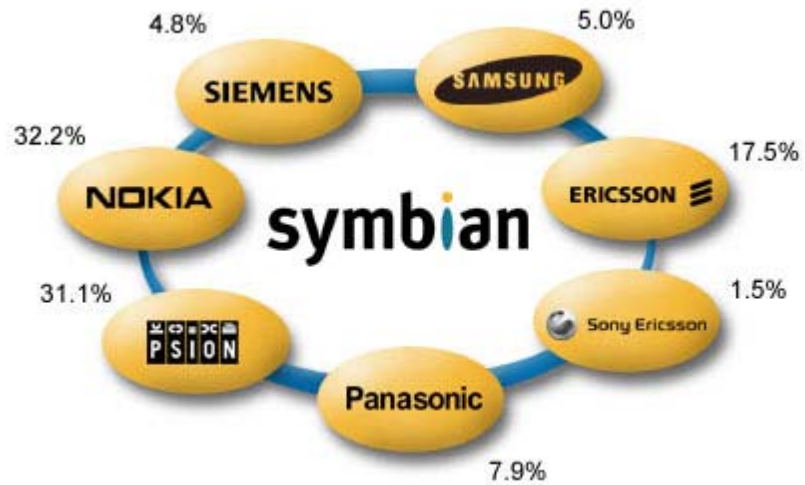
Tämä opinnäytetyö keskittyy tarkastelemaan älypuhelimissa toimivaa Symbian OS – käyttöjärjestelmää sen toiminnallisuuden osalta. Tarkemmin tutustutaan Nokian Series 60 käyttöliittymään, joka toimii Symbian OS käyttöjärjestelmän päällä. Matkapuhelinpalveluiden osalta työssä tutustutaan multimediaviesteihin. Lisäksi työssä käydään läpi ohjelmistotestaamisen periaatteet. Työn käytännön osassa on toteutettu sovellus multimediaviestien vastaanottamiseen ja käsittelemiseen uuden sukupolven matkapuhelimessa ja tarkastellaan sovellusta testauksen näkökulmasta.

2 SYMBIAN YHTIÖ

Psion kehitteli 1990-luvulla EPOC-käyttöjärjestelmää, joka toimi pienissä, langattomissa taskumikroissa. EPOC-käyttöjärjestelmä oli alusta asti suunniteltu toimimaan oliopohjaisesti resursseiltaan rajallisissa mobiililaitteissa. Se sisälsi Psionin aikaisempiin käyttöjärjestelmiin verrattuna myös enemmän viestintämahdollisuuksia. (Jipping 2002, 10.)

Samoihin aikoihin johtavat matkapuhelinvalmistajat etsivät uutta, edistyksellistä ja laajennettavaa käyttöjärjestelmää tulevan sukupolven matkaviestimiin. Psion oli myös kiinnostunut EPOC:n käyttömahdollisuuksista matkapuhelimissa. Vuonna 1998 Psion perusti yhdessä Ericssonin, Nokian ja Motorolan kanssa Symbian -nimisen yhtiön, jonka tarkoituksena oli ottaa EPOC-käyttöjärjestelmä omistukseensa ja ryhtyä kehittämään sitä seuraavien sukupolvien matkapuhelimiin sopivaksi. (Jipping 2002, 10.)

2002 vuoteen mennessä Symbian-yhtiöön liittyivät myös Panasonic, Samsung, Siemens ja Sony Ericsson. Syksyllä 2003 Nokia osti Motorolan Symbian-osakkeet ja suunnittelee nyt myös Psionin omistusosuuden ostamista. Tällöin Nokian osuus yhtiön omistuksesta nousi yli 60 prosenttiin. Kuviossa 1 on esitetty Symbian yhtiön omistajien prosenttiosuudet yhtiöstä. (FCS Partners; Sanoma Magazines Finland 2003.)



Kuvio 1. Symbianin omistussuhteet (Symbian Ltd 2004 a.)

Symbian yhtiö kehittää ja lisensoi edistyksellistä ja avointa käyttöjärjestelmää seuraavien sukupolvien langattomiin päätelaitteisiin. Yhtiön visio on 'Symbian käyttöjärjestelmä kaikkiin puhelimiin'. Tavoitteena on, että kaikki yhtiössä mukana olevat yritykset hyötyisivät yhteisestä käyttöjärjestelmästä. Kun markkinoilla on vain yksi yhteinen käyttöjärjestelmä, kaikki säästävät kehityskustannuksissa ja riskit pienenevät. (Symbian Ltd 2004 b; Helsingin Teknillinen Korkeakoulu 2002.)

3 SYMBIAN OS ÄLYPUHELINTEN KÄYTTÖJÄRJESTELMÄNÄ

Älypuhelimiksi kutsutaan kolmannen sukupolven matkapuhelimia, joissa tyypillisesti on grafiikan esittämisen mahdollistava värinäyttö ja kehittyneitä, lisäarvoa tuovia sovelluksia. Älypuhelin voidaan liittää verkon kautta esimerkiksi tietokoneeseen tai televisioon. Älypuhelimissa on myös kehittyneet viestintäominaisuudet, jotka mahdollistavat sähköposti- ja multimediamiestien lähettämisen ja vastaanottamisen.

3.1 Vaatimukset älypuhelimien käyttöjärjestelmälle

Matkapuhelin on laite, jota tuskin koskaan sammutetaan. Tämä asettaa vaatimuksia käyttöjärjestelmän luotettavuudelle. Järjestelmän täytyy toimia jatkuvasti ilman uudelleenkäynnistämistä ja sen täytyy olla vakaa ja tietojen pitää säilyä kaikissa tilanteissa. Käyttöjärjestelmä ei saa kuluttaa liian paljon muistia, koska pienen langattoman päätelaitteen resurssit ovat hyvin rajalliset. (Helsingin Teknillinen Korkeakoulu 2002.)

Jotta laite olisi mahdollisimman käyttäjäystävällinen, se ei saa myöskään olla suurikokoinen tai liian painava. Akun tulee olla pieni, mikä rajoittaa energiankulutuksen määrää. Käyttöjärjestelmän täytyy kuluttaa mahdollisimman vähän virtaa, jotta akkua ei tarvitsisi ladata turhan usein. Koska langattomia päätelaitteita on monia erilaisia, tulee käyttöjärjestelmän olla joustava ja sopia erilaisiin kokoonpanoihin. (Helsingin Teknillinen Korkeakoulu 2002.)

3.2 Symbian OS:n kuvaus

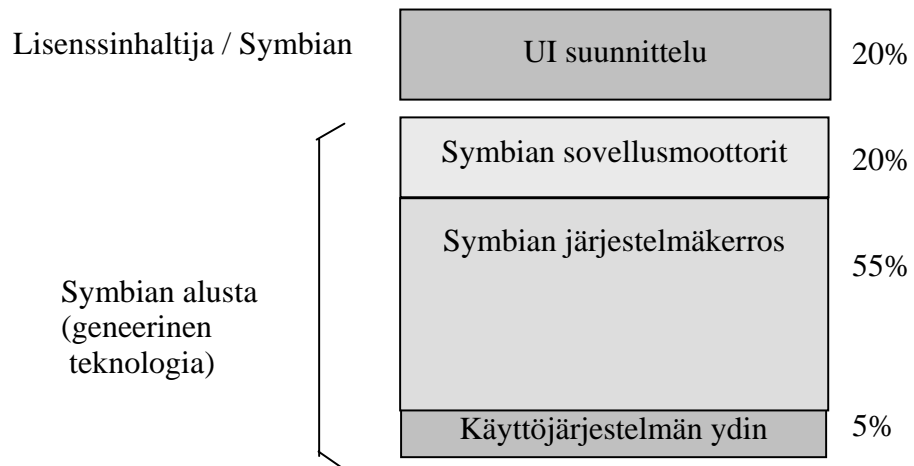
Symbian yhtiö on määritellyt viisi erityispiirrettä, jotka tekevät matkapuhelinmarkkinoista ainutlaatuisen ja luovat tarpeen uudelle käyttöjärjestelmälle:

- matkapuhelimet ovat sekä pieniä että langattomia
- matkapuhelimet ovat käytössä kaikkialla – kuluttajien massamarkkinoilla, yritys- ja ammattikäytössä
- matkapuhelinta käytetään langattomassa puheliverkossa, liitettynä toiseen laitteeseen tai itsenäisenä laitteena
- matkapuhelinvalmistajat haluavat erilaistaa tuotteitaan sen sijaan, että joutuisivat innovoimaan ja kilpailemaan nopeasti kehittyvillä markkinoilla
- järjestelmän täytyy olla avoin ja mahdollistaa ohjelmistokehittäjien oma sovelluskehitys, teknologiat ja palvelut.

Symbian käyttöjärjestelmä on suunniteltu ja kehitetty täyttämään nimenomaan langattomille päätelaitteille asetetut vaatimukset. Se on suunniteltu toimimaan akkukäyttöisissä ja muistikapasiteetiltaan rajallisissa laitteissa. Käyttöjärjestelmä mahdollistaa itsenäisen sovelluskehityksen Symbian alustalle. Järjestelmän käyttövarmuuden ja vakauden ansiosta sovellukset voivat olla käynnissä jopa vuosia tarvitsematta uudelleenkäynnistystä tai menettämättä käyttäjätietoja. Käyttöjärjestelmä on suunniteltu oliopohjaiseksi ja sen tarjoama C++ API sallii pääsyn mm. puhelintekniikkaan ja viestintään. Rakenteensa ansiosta Symbian sopii moniin eri päätelaitteisiin. (Symbian Ltd 2004 c; FCS Partners.)

3.2.1 Käyttöjärjestelmän rakenne

Kuviossa 2 on esitetty Symbian käyttöjärjestelmän nelikerroksinen rakenne. Suurin osa (80%) Symbianin ominaisuuksista on riippumattomia näytön koosta ja tiedon syöttämistavasta. Tämä osa on käyttöjärjestelmän ydin, joka on samanlainen kaikissa Symbian päätelaitteissa. Toinen osa kokonaisuudesta (20%) on varattu graafiselle käyttöliittymälle, jota voidaan muokata erilaisiin laitteisiin sopivaksi. (Symbian Ltd 2004 c.)



Kuvio 2. Symbian käyttöjärjestelmän rakenne (FCS Partners.)

Symbian käyttöjärjestelmä toimii älypuhelimissa, kommunikaattoreissa ja taskumikroissa. Älypuhelimissa tietoa syötetään puhelimen painikkeiden avulla, kommunikaattoreissa näppäimistön kautta ja taskumikroissa koskettamalla näyttöä erityisellä kynällä. (Symbian Ltd 2004 c.)

3.2.2 Symbian OS 7.0s version toiminnallisuus

Symbian OS 7.0s tarjoaa toiminnallisuuksiltaan kolmannen sukupolven älypuhelinmarkkinoille sopivan alustan ja mahdollistaa näin 3G palvelujen välittämisen (Symbian Ltd 2004 d). 7.0s version tärkeimpiä uusia ominaisuuksia ovat:

- tuki kaikille uusimmille langattomille puhelintekniikoille
- tehokkaampi ja laajempi multimediatekniikka, joka korvaa aikaisempien versioiden multimediatekniikan
- Java MIDP 2.0 tuki erityisesti peleihin
- SyncML DS standardi
- kaksisuuntainen tuki arabian ja heprean kielille.

Puhelintekniikka ja verkkoyhteydet

Symbian OS 7.0s tukee mm. uutta WCDMA laajakaistaista radioliikennetekniikkaa, joka tarjoaa runsaasti kapasiteettia internet-, multimedia- ja videosovelluksiin (Telefonaktiebolaget L M Ericsson. 1994 – 2003). Aikaisempien Symbian versioiden verkkoarkkitehtuuri mahdollisti vain yhden verkkoyhteyden kerrallaan, mutta 7.0s versiossa on mahdollista luoda useita samanaikaisia erilaista verkkokapasiteettia vaativia yhteyksiä. Esimerkiksi puhelun aikana on mahdollista vastaanottaa multimediamiestejä ja samaan aikaan ajaa taustalla kalenterin synkronointia PC:n kanssa. (Nokia 2004 a.)

Multimediakehys

Uusi MMF-multimediakirjasto on suunniteltu erityisesti kuvaa ja ääntä yhdistäville sovellusalueille. Se mahdollistaa videoiden ja muiden suurta kaistanleveyttä vaativien sovellusten käsittelyn. MMF poistaa resurssiongelmat, joita voi esiintyä mm. yhtäaikaista soundtrackeja vaativissa sovelluksissa kuten peleissä. (Nokia 2004 a.)

Java MIDP 2.0

Java MIDP 2.0 on sovelluskehitysympäristö pääasiassa älypuhelimien ja kaksisuuntaisten hakulaitteiden ohjelmointiin (Arokoski, Jääskeläinen, Kontio, Köykkä, Raatikainen, Tervo & Vierimaa 2002, s. 145). Symbian OS 7.0s sisältää tuen MIDP 2.0 sovelluksille, joita voidaan ladata internetistä puhelimeen.

SyncML tuki

SyncML mahdollistaa esimerkiksi kannettavassa tietokoneessa ja matkapuhelimessa olevien tietojen yhteensovittamisen tietoverkon kautta. Aikaisemmin tietojen synkronointi on ollut rajoitettua erilaisten sovelluskohtaisten protokollien takia. SyncML tarjoaa standardin, jonka avulla tietojen synkronointi tietoverkkojen, ohjelmistoalustojen ja laitteiden välillä onnistuu. (Nokia 2004 b.)

4 SERIES SARJAN OHJELMISTOKEHITYSALUSTAT

Series sarjan ohjelmistokehitysalustat ovat Nokian kehittämiä ja lisensoimia avoimia ohjelmistokehitysalustoja. Niiden tarkoituksena on tarjota ohjelmistokehittäjille uusia liiketoimintamalleja ja pääsy massamarkkinoille. (Nokia 2004 f.)

4.1 Series 40

Series 40 on Nokian ensimmäinen käyttöliittymä, joka tukee värinäytöllisiä matkapuhelimia ja johon voidaan kehittää omia sovelluksia. Ohjelmistoalusta toteuttaa kolmea erilaista käyttöliittymämallia, joista yleisin on 128 x 128 pikselin näytölle. Näppäimistö sisältää numeeristen näppäinten lisäksi kaksi- tai nelisuuntaisen navigointipainikkeen sekä kaksi tai kolme näytöllä näkyvää valonäppäintä (softkey). (Nokia 2004 e.)

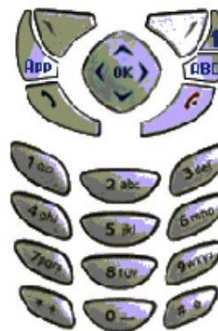
Series 40 sarjan puhelimet sisältävät integroidun kameran, tuen langattoman viestinnän API:lle (JSR-120) ja selailuun XHTML:lle. Series 40 sarjan puhelimet hyödyntävät Java-teknologiaa. Nokian 7210 matkapuhelin on suunniteltu Series 40 sarjan viitetoteutus malliksi. (Nokia 2004 e.)

4.2 Series 60

Series 60 ohjelmistokehitysalusta on Nokian ensimmäinen avoin ohjelmistokehitysalusta Symbian älypuhelimiin. Se sisältää yleisen käyttöliittymän yhdellä kädellä käytettäville älypuhelimille. Series 60 ohjelmistoalustalle voidaan tehdä sekä ns. stand-alone sovelluksia, että palvelinsovelluksia. Myös erillisiä Java ja WAP/XHTML työkaluja voidaan käyttää sovellusten suunnittelussa. Ohjelmistokehittäjillä, operaattoreilla ja laitevalmistajilla on mahdollista kehittää omia palvelinsovelluksiaan alustalle: esimerkiksi sovellus, jolla yrityksen tietoverkon etäkäyttäjät voidaan liittää verkkoon langattomasti. (Nokia 2004 d.)

4.2.1 Käyttöliittymä

Series 60 ohjelmistoalusta tukee 176 x 208 pikselin näyttöjä ja 12 painikkeen numeerista näppäimistöä. Toimintonäppäimiä ovat kaksi valopainiketta (softkey), nelisuuntainen navigointipainike, sovellusten käynnistys- ja lopetuspainikkeet, vihreä ja punainen luurinäppäin sekä tekstin muokkaamiseen tyhjennys- ja aakkosnäppäin. Kuviossa 3 on esimerkki Series 60 ohjelmistoalustan tukemasta näppäimistöstä. (Nokia 2004 d.)



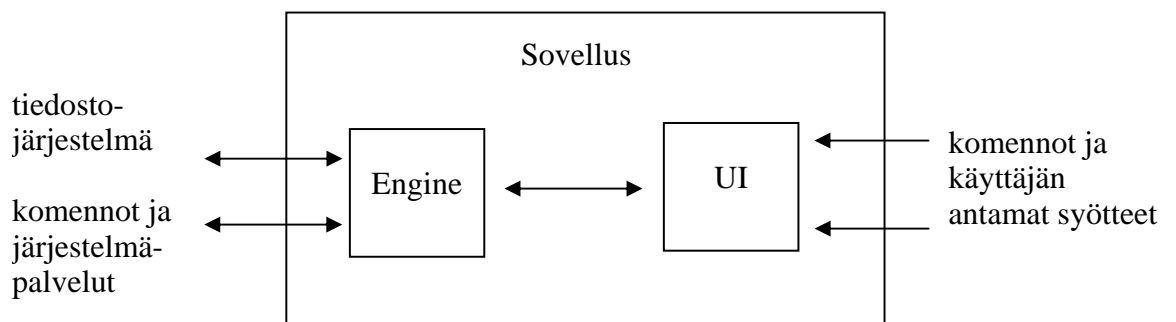
Kuvio 3. Series 60 ohjelmistoalustan tukema näppäimistö

Series 60 alusta tarjoaa runsaasti PC ohjelmien kehittäjille tuttuja käyttöliittymäkomponentteja, kuten listbox, dialogit, pop-up menut, checkbox, radiopainikkeet ym. (Nokia 2004 d.)

4.2.2 Sovellusten perusrakenne

Series 60 ohjelmistoalustan sovellukset voidaan jakaa kahteen osaan: käyttöliittymään ja sovellusmoottoriosaan (engine). Sovellusmoottori käsittelee sovelluksen tiedot ja muut käyttöliittymästä riippumattomat toiminnot. Sovellusmoottori ei ole pelkästään yhden tietyn sovelluksen käytössä, vaan sitä voi käyttää useampi eri sovellus. Käyttöliittymäosa näyttää tiedot käyttäjälle. (FCS Partners.)

Kuviossa 4 on esitetty Series 60 ohjelmistoalustan sovellusten perusrakenne. Käyttöliittymään (UI) tulee komentoja sovelluksen ulkopuolelta tai käyttäjän antamia syötteitä. Käyttöliittymä kommunikoi sovellusmoottorin kanssa (engine), joka käsittelee tiedot ja lähettää ne edelleen järjestelmän alemmille tasoille sekä palauttaa tietoa myös käyttöliittymälle. (FCS Partners.)

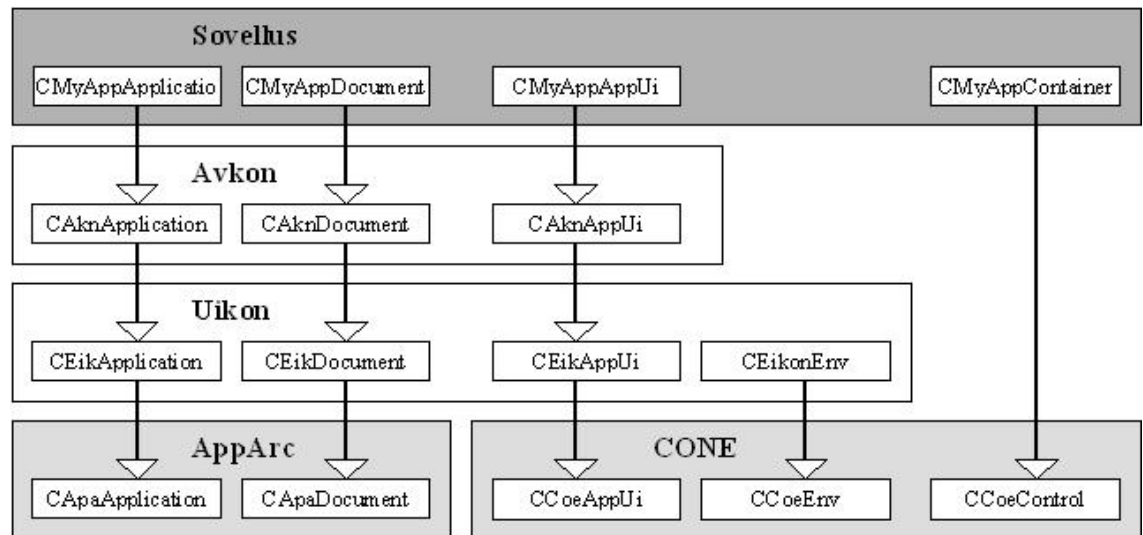


Kuvio 4. Series 60 sovellusten perusrakenne

Series 60 sovellusten luokkien suhteet Symbian OS käyttöjärjestelmän kanssa on esitetty kuviossa 5. Sovelluskerroksella on yksinkertainen MyApp –niminen sovellus ja sen perusluokat. (FCS Partners.)

Avkon kerrokselle sijoittuvat Symbian OS käyttöjärjestelmän päällä toimivat käyttöliittymämallit, joka tässä kuviossa on Series 60. Tämä kerros sisältää ohjelmistoalustalle spesifiset luokat (CAkn luokat), joissa toteutetaan Series 60 käyttöliittymämallin toiminnallisuus ja rajoitukset verrattuna Uikon kerrokseen. Uikon on kaikille Symbian OS käyttöjärjestelmän päällä toimiville käyttöliittymämalleille yhteinen käyttöliittymä ja kontrollisovelluskehys. (FCS Partners.)

AppArc kerros (CApa luokat) määrittelee rajapinnan sovelluksen luokille ja prosesseille (ei ole esitetty kuviossa). CONE kerros määrittelee perustason kontrolliluokan (CCoeControl) ja lisäksi alemman tason luokat asynkroniselle tapahtumankäsittelylle (CCoeAppUi ja CCoeEnv). (FCS Partners.)



Kuvio 5. Luokkien johtaminen Series 60:ssä

4.3 Series 90

Series 90 ohjelmistoalustaa käytetään kommunikaattoreissa ja se pohjautuu Symbianin versioon 7.0s. Se täydentää aikaisempia series sarjan ohjelmistoalustoja ja on paranneltu kommunikaattoriversio Series 80:sta. (Nokia 2004 g.)

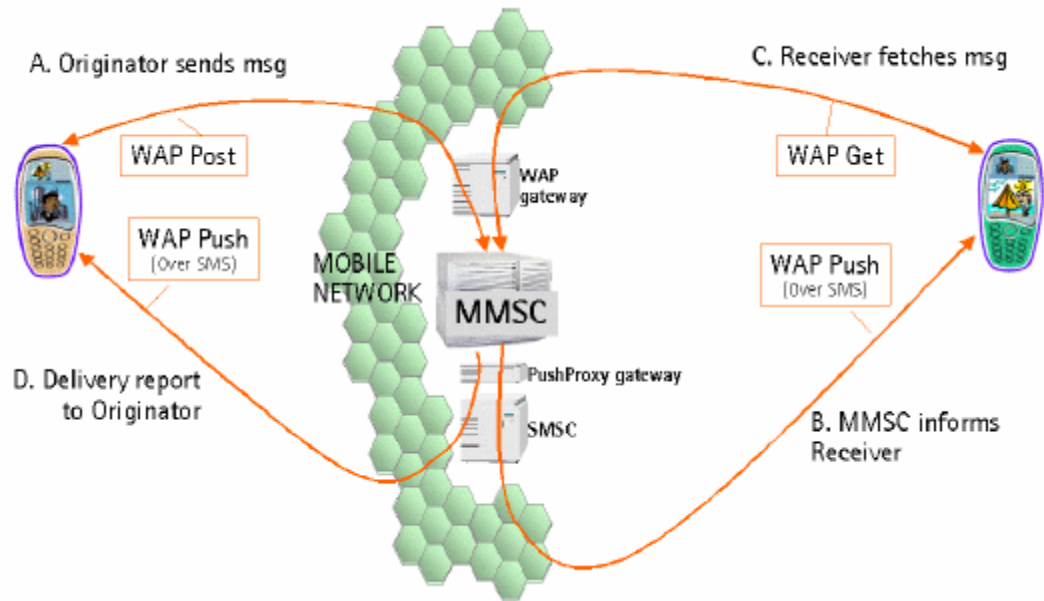
Käyttöliittymä sisältää 640 x 320 pikselin ja 65536 värin kosketusnäytön. Laitteissa voidaan käyttää perinteisen kommunikaattorinäppäimistön lisäksi myös kämmenmikroista tuttua kynää. Sen avulla käyttöliittymä tunnistaa käsin kirjoitetun tekstin. Käyttäjä voi halutessaan vaihtaa käyttöliittymän graafista ulkoasua valitsemalla eri teemoista, joita käyttöliittymässä on valmiina. Series 90 alustalle voidaan kehittää sovelluksia C++ ja Java kielillä tai web-pohjaisia sovelluksia käyttäen HTML, XHTML tai Macromedia Flash standardeja. (Nokia 2004 h.)

5 MULTIMEDIAVIESTINTÄ

Multimedia messaging service (MMS) on WAP-Forum ja 3GPP:n standardisoima tekstiviestin kaltainen viestintäpalvelu, jolla voidaan lähettää kuvia, ääntä, animaatioita ja videoleikkeitä ensisijaisesti matkapuhelimesta toiseen. Toisin kuin tekstiviesteissä, multimediaviestin kokoa ei ole erityisesti rajattu. MMS-viestejä voi lähettää myös matkapuhelimesta sähköpostiin tai päinvastoin. Viestien siirtämiseen käytetään WAP-protokollia. Matkapuhelinverkon ulkopuolelta lähetetyt viestit välitetään multimediaviestikeskuksen kautta. (Tampereen Teknillinen yliopisto 2003 a; Nokia 2004 c.)

5.1 MMS viestien kulku tietoverkossa

Multimediaviestien lähettäminen toimii periaatteessa samalla tavalla kuin tekstiviestien lähettäminen. MMS viestien lähettäminen onnistuu, kun päätelaitteeseen on määritelty MMS viestipalveluasetukset (mm. MMSC viestipalvelimen osoite) ja kun päätelaite mahdollistaa WAP GPRS yhteyden luomisen. Tarkemmin MMS viestien kulku tietoverkossa on esitetty kuviossa 6. (Nokia 2004 i.)

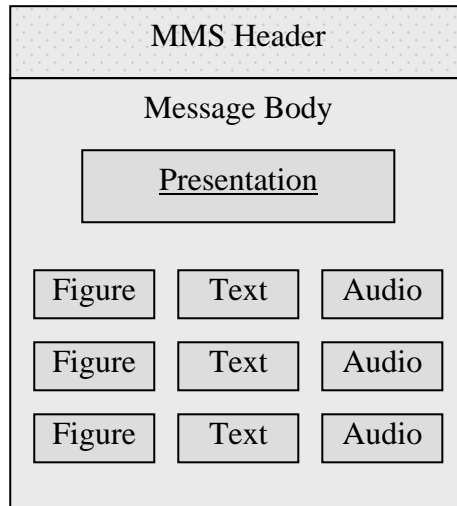


Kuvio 6. Multimediaviestien kulku tietoverkossa (Nokia 2004 i.)

Viesti lähetetään toisesta matkapuhelimesta (originator) vastaanottavaan (receiver) matkapuhelimeen. Viesti kulkee WAP yhdyskäytävän (WAP gateway) kautta ensin MMSC viestipalvelimelle. Viestikeskus lähettää (B) ilmoituksen uudesta multimediaviestistä vastaanottajalle tekstiviestinä PushProxy yhdyskäytävän ja tekstiviestipalvelimen (SMSC) kautta. Kun vastaanottavan matkapuhelimen käyttäjä haluaa avata viestin, se haetaan (C) viestipalvelimelta. Tämän jälkeen viestipalvelin lähettää ilmoituksen (D) viestin lähettäneeseen matkapuhelimeen. (Nokia 2004 i.)

5.2 MMS viestin rakenne

Multimediaviestien rakennetta ei ole määritelty WAP standardissa. Multimediaviestit koostuu multimediaesitystä kontrolloivista script käskyistä ja multimediaa sisältävistä objekteista (esim. kuva, teksti tai ääniobjektit). Viestissä on kaksi osaa, joista header osa sisältää tietoja mm. viestin lähettäjistä, vastaanottajasta ja viestin sisältötyypistä. Body osa sisältää varsinaisen viestin datan. Multimediaviestit ovat multimediaesityksiä, joissa käytetään SMIL kuvauskieltä internetissä käytetyn HTTP kuvauskielen tapaan. Kuviossa 7 on esimerkki multimediaviestin rakenteesta. (Symbian DevNet 2002.)



Kuvio 7. Multimediaviestin rakenne (Nokia 2004 i.)

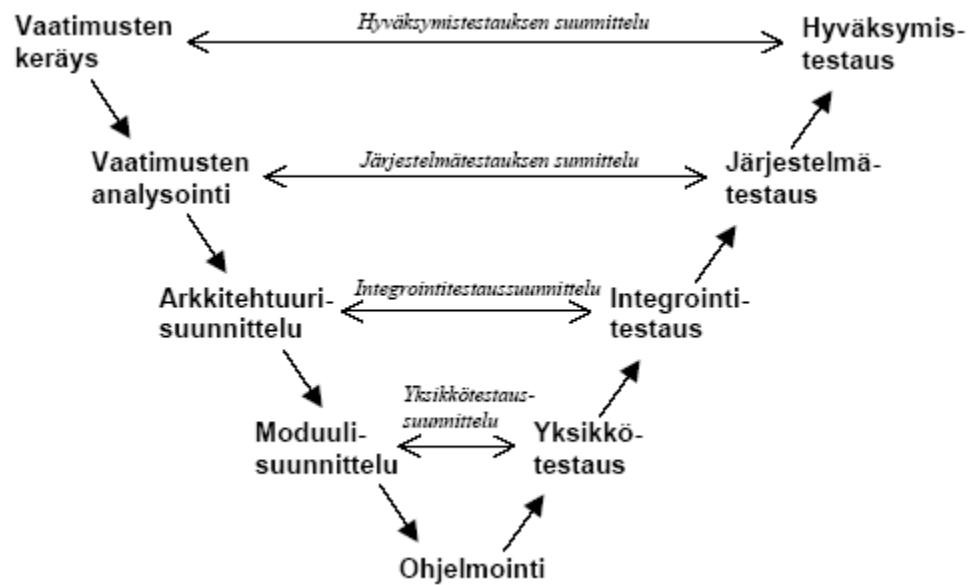
SMIL kuvauskielen avulla voidaan mm. määritellä esitettävien multimediaobjektien layout ja ajastaa eri objektien näkyminen eri aikaan. SMIL esitys voidaan myös ohittaa ohjelmallisesti ja toteuttaa objektien näyttämisen omalla tavalla. (Symbian DevNet 2002.)

6 MOBIILI PÄÄTELAITTEEN OHJELMISTOTESTAAMINEN

Mobiilipäätelaitteiden ohjelmistojen testaaminen on kallista ja hankalaa toteuttaa pelkästään todellisessa ympäristössä. Tämän vuoksi testaamiseen on kehitetty runsaasti eri alustoille sopivia PC-pohjaisia emulaattoreita, joilla sovelluksia voidaan testata oikeankaltaisessa ympäristössä tietokoneella. Emulaattoria voidaan käyttää yksikkö- ja integrointitestauksessa, mutta järjestelmä- ja hyväksymistestaaminen tehdään todellisessa ympäristössä kenttäolosuhteissa. (Tampereen Teknillinen yliopisto 2003 b.)

6.1 Testaamisen tasot

Ohjelmistotestaaminen jaetaan yleisesti neljään osaan ns. V-mallin mukaisesti (kuvio 8). Mitä pienemmissä osissa testaamista suoritetaan, sitä helpommin virheet voidaan havaita. V-malli osoittaa myös, mihin ohjelmistotuotantoprosessin vaiheeseen kyseisen testausvaiheen suunnittelu kuuluu. Testaaminen on siis oikein suunniteltuna ja toteutettuna kiinteä osa ohjelmistokehitysprosessia. (Satukangas, A. 2003.)



Kuvio 8. Testaamisen V-malli (Satukangas, A. 2003.)

Yksikkö- ja integrointitestaus ovat rakenteellista testaamista (white-box testing), kun taas järjestelmä- ja hyväksymistestaus ovat toiminnallista testaamista (black-box testing). Rakenteellisessa testaamisessa lähdekoodia tarkastellaan pieninä osina ja todennetaan, että ohjelmakoodi antaa oikean lopputuloksen tietyllä suorituspolulla. Toiminnallisessa testaamisessa puolestaan testataan ohjelman ulkoista toiminnallisuutta ja käytettävyyttä. (Satukangas, A. 2003.)

6.1.1 Yksikkötestaus

Yksikkö- eli moduulitestauksessa testataan ohjelman yksittäistä riippumatonta osaa, moduulia, joka koostuu yleensä noin 100-1000 koodirivistä. Testaamisen suorittaa usein moduulin tekijä, mutta tällöin myös virheiden löytymisen todennäköisyys on pienin. Moduulien testaamiseen tarvitaan usein testiajureita, jotka korvaavat yksikköä kutsuvan ohjelman, ja tynkämoduuleita, jotka korvaavat muut yksikön tarvitsemat moduulit. (Satukangas, A. 2003; Haikala, I. & Märijärvi, J. 2002, 287.)

6.1.2 Integrintitestaus

Integrintitestauksessa yhdistellään moduuleita ja testataan niiden välisiä rajapintoja. Testauksessa tutkitaan komponenttirajapintojen välistä toiminnallisuutta ja sitä, että tieto niiden välillä kulkee oikein. Integrintitestausta voidaan suorittaa rinnakkain yksikkötestauksen kanssa aina silloin, kun uusia moduuleita syntyy. (Satukangas, A. 2003; Haikala, I. & Märijärvi, J. 2002, 287-288.)

Moduulien integroimisessa käytetään joko kokoavaa eli ylhäältä alas- (top-down) tai jäsentävää eli alhaalta ylös -strategiaa (bottom-up). Kokoavassa integroinnissa komponentteja voidaan kehittää rinnakkain ja alemman tason komponenttien valmistamista voidaan tarvittaessa lykätä. Jäsentävä integrointi puolestaan mahdollistaa moduulien rinnakkaisen kehittämisen ja testaamisen. (Satukangas, A. 2003.)

6.1.3 Järjestelmä- ja hyväksymistestaus

Järjestelmätestausvaiheessa testataan järjestelmän toimivuutta kokonaisuutena. Testauksen suorittavat kehitystyön ulkopuolella olevat henkilöt, jotta liiallinen tietämys oikeanlaisesta toiminnallisuudesta ei häiritse testaamista. Tässä vaiheessa ilmenevät virheet voivat johtua esimerkiksi puuttuvasta toiminnosta tai toiminnallisuudesta, joka ilmenee vasta testattaessa kenttäolosuhteissa. Ulkoisen toiminnallisuuden lisäksi järjestelmätestausvaiheessa voidaan suorittaa stressi-, kuormitus-, suorituskyky-, toipumis- sekä tietoturvaluottelutestauksia. (Satukangas, A. 2003; Haikala, I. & Märijärvi, J. 2002, 288.)

Hyväksymistestaus on hyvin samankaltainen kuin järjestelmätestausvaihe, mutta sen suorittaa lopullinen asiakas ja virheet liittyvät usein ohjelman käytettävyyteen. Järjestelmä- ja hyväksymistestausvaiheet voidaan usein yhdistää. (Satukangas, A. 2003.)

6.2 Emulaattori

Kehitettäessä sovelluksia mobiilipäätelaitteisiin, ne testataan ensin PC-pohjaisella emulaattorilla ennen varsinaista järjestelmä- ja hyväksymistestausta. Symbian alustalle kehitettyjä emulaattoreita voidaan käyttää Windows ympäristössä. Esimerkiksi Microsoftin Visual Studio tai Borlandin C++ ohjelmistokehitysympäristö voidaan yhdistää emulaattoriin, jolloin sovelluksia voidaan debugata paremmin. Kuviossa 9 on esimerkki Series 60 ohjelmistoalustan emulaattorista. (FCS Partners.)



Kuvio 9. Nokian Series60 emulaattori

Windows-pohjainen emulaattori mahdollistaa Symbian sovellusten kirjoittamisen Windows ympäristössä. Debuggaustyökalun lisäksi emulaattori tarkistaa muistinkulutusta

debuggauksen aikana. Emulaattori ilmoittaa muistivuodoista, jos niitä ilmenee sovelluksesta poistuttaessa. (FCS Partners.)

Emulaattori toimii oikean matkapuhelimen kaltaisesti, mutta ilman yhteyttä tietoverkkoon. Sen painikkeita voidaan käyttää tietokoneen näppäimistön tai hiiren avulla. Multimediaviestien testaamista varten PC:lle on mahdollista asentaa MMS viestipalvelinta mallintava työkalu, jolla voidaan lähettää multimediaviestejä tietokoneessa oleville erilaisille emulaattoreille.

7 SOVELLUS MULTIMEDIAVIESTIEN VASTAANOTTAMISEEN

Käytännön osassa oli tarkoitus tutustua Symbian ohjelmointiin toteuttamalla sovellus multimediaviestien käsittelyä varten ja testata sitä ohjelmistotestaamisen näkökulmasta. Sovellus toteutettiin projektityönä laaditun projektisuunnitelman ja vaatimusmäärittelyn mukaan.

7.1 Tavoitteet

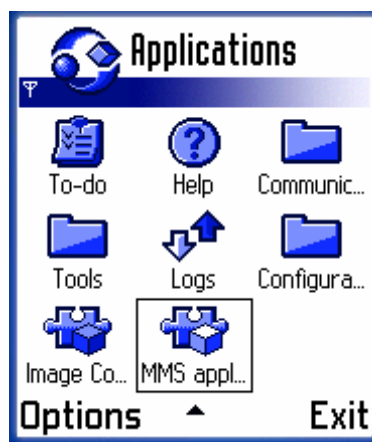
Opinnäytetyöprojektin käytännön osassa tavoitteena oli tehdä sovellus, joka on tarkoitettu multimediaviestien (MMS) vastaanottamiseen ja käsittelemiseen Symbian OS käyttöjärjestelmän ja Series 60 ohjelmistoalustan sisältävissä päätelaitteissa. Sovelluksen tuli sisältää seuraavat toiminnot:

- viestin automaattinen noutaminen MMSC palvelimelta sen jälkeen, kun päätelaitteeseen on saapunut ilmoitus uudesta viestistä
- tietyllä etuliitteellä (/EB) saapuvien viestien poimiminen viestivirrasta
- ilmoitusviestin näyttäminen käyttöliittymässä
- viestin avaaminen tai poistaminen.

Sovelluksen toiminnallisuus määriteltiin opinnäytetyön aloituspalaverissa. Sen pohjalta tehtiin vaatimusmäärittely, jonka tarkoituksena oli kuvailla opinnäytetyöprojektissa tehtävä

ohjelmisto mahdollisimman tarkasti ja asettaa sille toiminnallisia ja ei-toiminnallisia vaatimuksia. Dokumentti oli sopimus tehtävästä työstä toimeksiantajan ja projektin välillä.

Sovelluksen tärkein ominaisuus on multimediatekstien vastaanottaminen. Ominaisuus kuvaillaan vaatimusmäärittelyssä seuraavasti: ”Sovellus käynnistetään erillisestä kuvakkeesta (esimerkki kuviossa 10), jonka jälkeen se voi reagoida viestipalvelimelta tuleviin ilmoituksiin (WAP push). Ilmoituksen saavuttua sovellus avaa WAP-yhteyden, noutaa viestin MMSC –palvelimelta (WAP get) ja tallentaa sen laitteen RAM-muistiin. Sovellus ilmoittaa käyttäjälle viestin saapumisesta.”



Kuvio 10. Series 60 älypuhelimien päävalikko

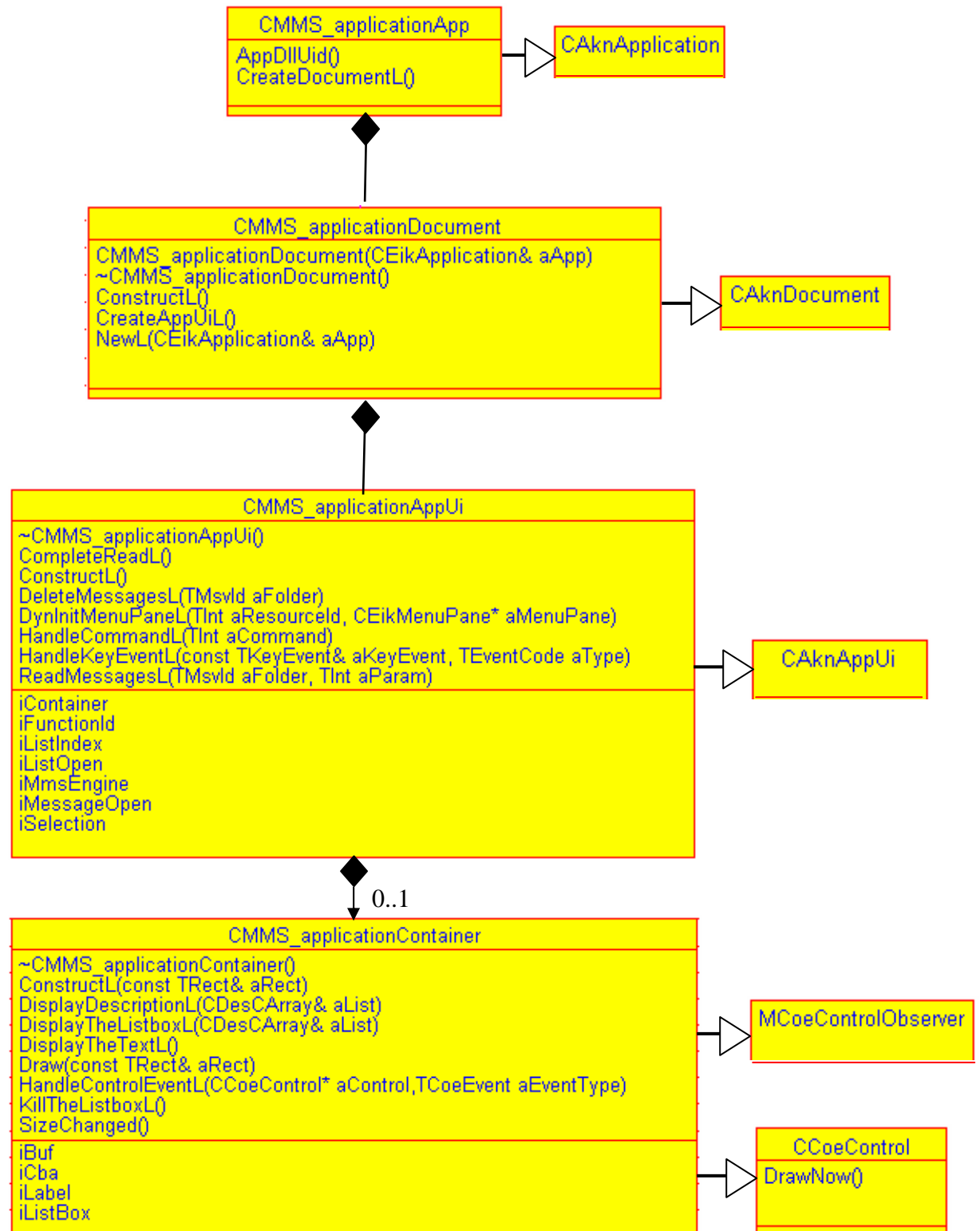
Käyttöliittymän ominaisuuksista vaatimusmäärittelyssä mainitaan seuraavaa: ”MMS viestin saavuttua päätelaitteeseen, sen näytöllä näkyy viesti ”Multimedia message received” ja vasen valonäppäin (softkey) on options. Options –painikkeesta avautuu valikko, jossa on vaihtoehdot open, remove ja exit.”

7.2 Suunnittelu

MMS sovellus käyttää perinteistä Symbian OS sovellusarkkitehtuuria. Projektissa ei ollut tarkoituksena luoda kaikkea toiminnallisuutta itse, joten sovellukseen on lisätty sovellusmoottori ja muutamia muita funktioita Forum Nokian internet sivustolta löytyneestä MMSSDemo esimerkistä (Nokia 2004 j).

7.2.1 Luokkakaaviot

Kuviossa 11 on kuvattu MMS sovelluksen luokkarakenne ja luokkien väliset suhteet.



Kuvio 11. MMS sovelluksen luokkarakenne

MMS sovellus koostuu Symbian sovellusten perusarkkitehtuurin mukaisesti seuraavista luokista:

CMMS_applicationApp

CMMS_applicationApp määrittelee MMS sovelluksen ominaisuudet ja toimii sovelluskehityksen käynnistysluokkana. Tätä luokkaa kutsutaan ensimmäisenä, kun sovellus käynnistetään päävalikosta. Luokka periytyy Avkon kerroksen luokasta *CAknApplication*.

CMMS_applicationDocument

Dokumenttiluokka huolehtii pysyvän tiedon hausta ja talletuksesta. Dokumenttiluokkia voi olla sovelluksessa vain yksi ja sovelluksen pääohjelma *CMMS_applicationApp* huolehtii sen alustamisesta. Tämä Avkon kerroksen luokasta *CAknDocument* periytyvä luokka täytyy olla jokaisessa sovelluksessa, vaikka tiedon lukemista levytä ja tallentamista ei tarvittaisikaan.

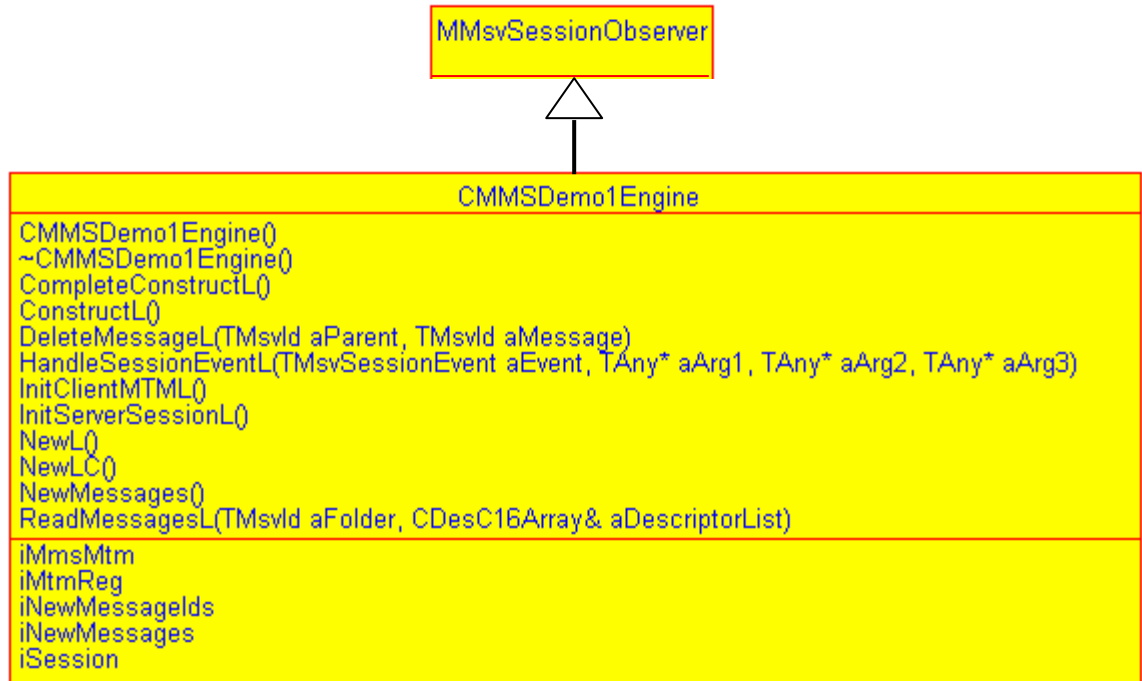
CMMS_applicationAppUi

CMMS_applicationAppUi on MMS sovelluksen käyttöliittymäluokka ja periytyy Avkon kerroksen luokasta *CAknAppUi*. *CMMS_applicationAppUi* sisältää sovelluksen tapahtumankäsittelyn ja käyttöliittymäkomponenttien toiminnallisuuden. Myös käyttöliittymäluokkia voi olla sovelluksessa vain yksi. *DeleteMessagesL()* ja *ReadMessagesL()* -funktiot on lisätty tähän luokkaan MMSDemo esimerkistä.

CMMS_applicationContainer

Container luokka sisältää käyttäjälle näkyvät komponentit ja toiminnallisuuden näytölle piirtämiseen. Jos sovellus käyttää view -arkkitehtuuria, jokaista view -luokkaa kohti täytyy olla oma Container luokka. MMS sovelluksessa Container luokkia on yksi. *CMMS_applicationContainer* -luokka sisältää label ja listbox kontrollit sekä toiminnallisuuden näiden luomiseen ja piirtämiseen näytölle.

DisplayTheListboxL() ja *KillTheListboxL()* –funktiot on lisätty tähän luokkaan *MMSDemo* esimerkistä. *DisplayDescriptionL()* ja *DisplayTheTextL()* –funktiot on luotu itse. *Container* luokka periytyy Avkon kerroksen *CCoeControl* ja *MCoeControlObserver* luokista.

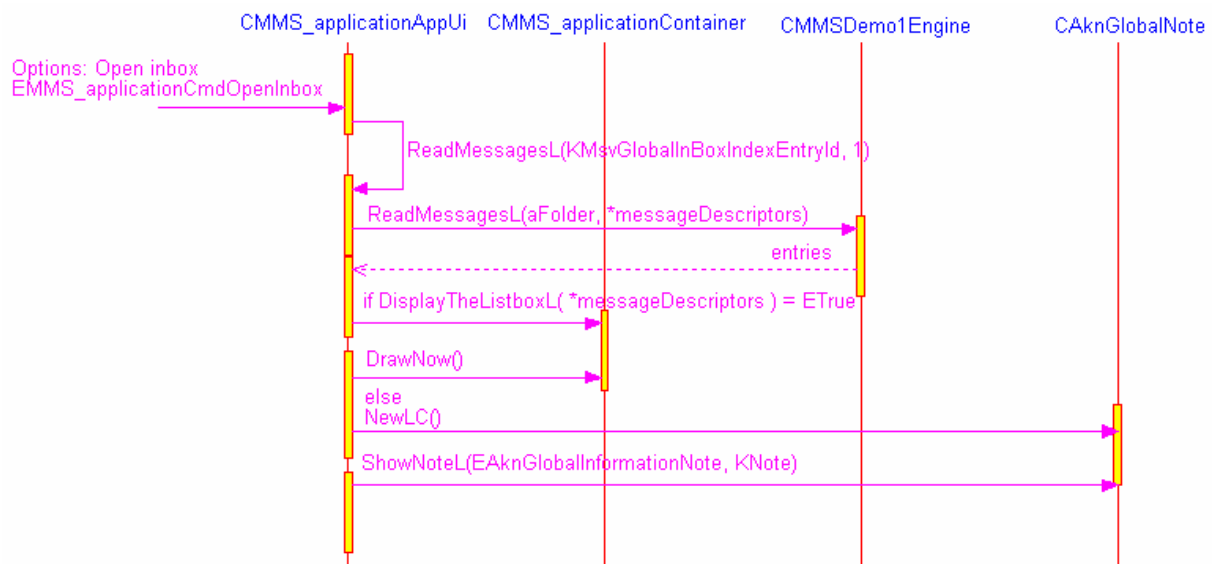


Kuvio 12. MMS sovellukseen liitetty sovellusmoottori

Kuviossa 12 on luokkakuvaus *CMMSDemo1Engine* sovellusmoottorista, joka on liitetty MMS sovellukseen Forum Nokian MMS Demo esimerkisovelluksesta. Luokasta on karsittu pois funktioita, joita ei tarvita tässä sovelluksessa. *CMMSDemo1Engine* periytyy *MmsvSessionObserver* –luokasta ja toimii vuorovaikutuksessa MMS palvelimen kanssa *CMmsClientMtm* asiakasrajapinnan kautta. Viestipalvelinyhteyksien luomisen lisäksi sovellusmoottori sisältää toiminnallisuuden MMS viestien vastaanottamiseen ja poistamiseen.

7.2.2 Sekvenssikaaviot

Sekvenssikaaviot kuvioissa 13-16 mallintavat MMS sovelluksen dynaamista käyttäytymistä eri käyttötapauksissa.



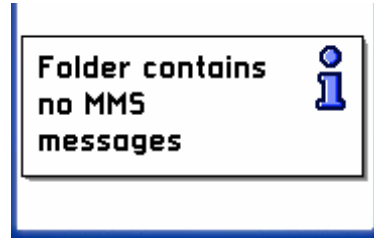
Kuvio 13. Sekvenssikaavio inboxin avaamisesta

Kuviossa 13 mallinnetaan tilannetta, jossa saapuneiden viestien inbox avataan. Tapahtumaan osallistuvat käyttöliittymäluokka *CMMS_applicationAppUi* ja komponentit sisältävä *CMMS_applicationContainer* luokka sekä sovellusmoottori *CMMSDemo1Engine*. Sovellus on käynnistetty päätelaitteen päävalikosta ja käyttäjä valitsee options-valikosta Open inbox -vaihtoehdon. Kyseisen valikkovaihtoehdon komento on *EMMS_applicationCmdOpenInbox* ja sen käsittely tapahtuu käyttöliittymäluokassa.

Käyttöliittymäluokan *ReadMessagesL* -funktiossa kutsutaan sovellusmoottorin vastaavaa funktiota, joka lukee viestit viestipalvelimelta globaalista inbox -kansioista taulukkoon. Viesteistä luetaan lähettäjän puhelinnumero ja varsinainen sisältö. Nämä tiedot käyttöliittymä lähettää edelleen *CMMS_applicationContainer* luokan *DisplayTheListboxL* -funktioille, joka tutkii, sisältääkö taulukko dataa ja luo listbox kontrollin valmiiksi.

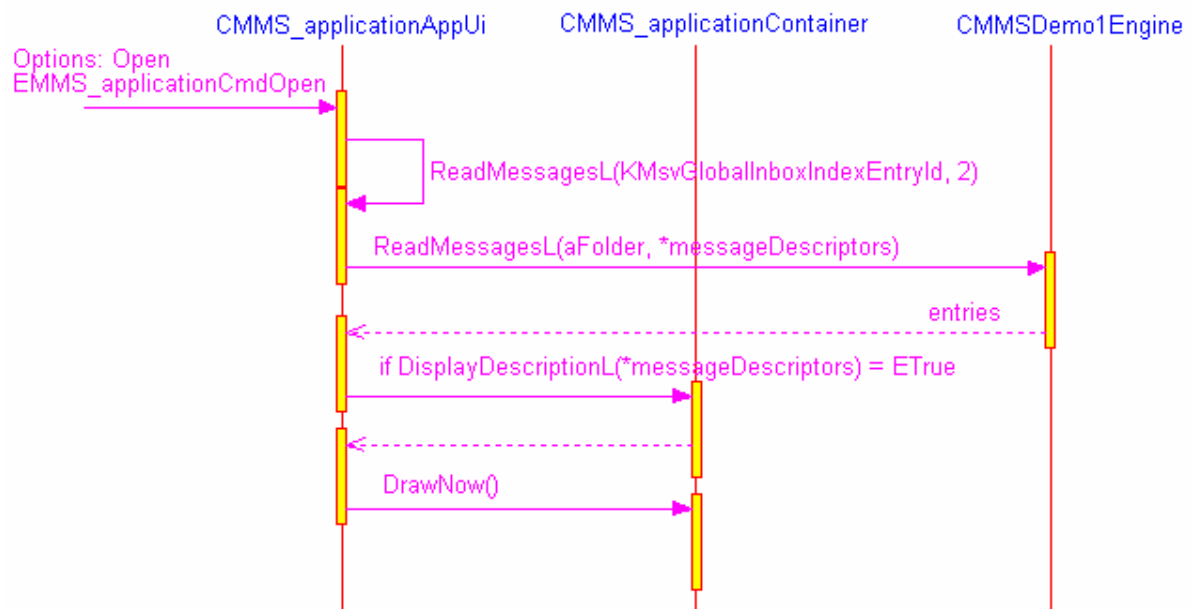
Jos *DisplayTheListboxL* -funktio palauttaa arvon *ETrue*, funktiossa luotu listbox kontrolli päivitetään ja näytetään siinä lähettäjän puhelinnumero ja alkuosa viestin tekstisisällöstä. Kontrollin tietojen päivittämiseen ohjelman ajon aikana käytetään *CMMS_applicationContainer* -luokan *CCoeControl* -luokasta perimää funktiota *DrawNow()*. Se kutsuu edelleen *CMMS_applicationContainer* -luokan funktiota *Draw*, joka suorittaa varsinaisen piirtämisen.

Jos taulukko on tyhjä, luodaan uusi *CAknGlobalNote* -tyyppinen olio ja kutsutaan sen funktiota *ShowNoteL*, joka saa parametrina ilmoituksen tyyppin ja tekstin. Kuviossa 14 on ilmoitusviesti, jonka MMS sovellus näyttää käyttöliittymässä inboxin ollessa tyhjä.



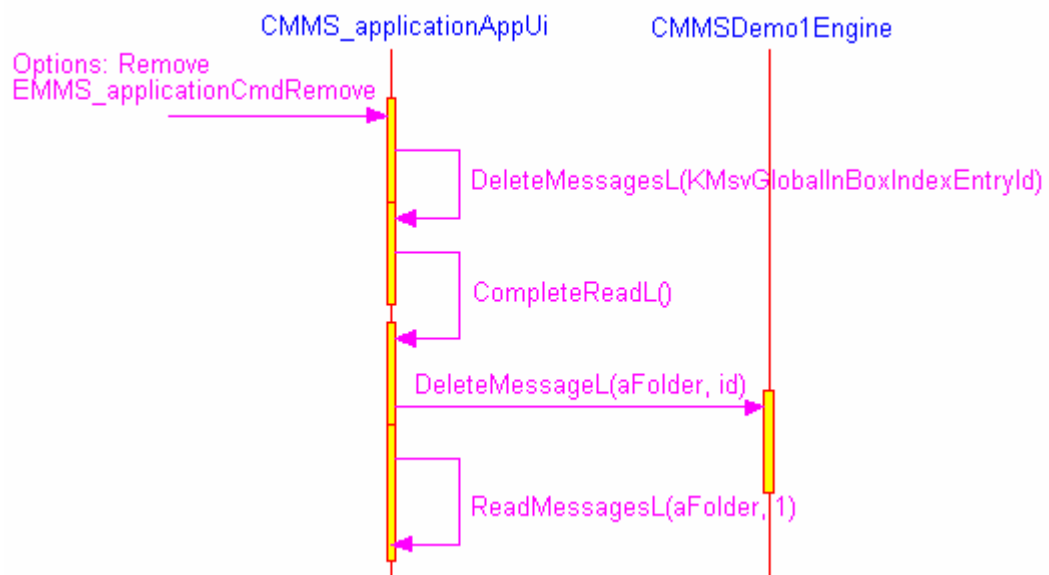
Kuvio 14. Näytöllä näkyvä ilmoitusviesti, kun inbox on tyhjä

Kuviossa 15 mallinnetaan viestin avaamista. Viesti avataan inboxista valitsemalla avattava viesti listalta ja valisemalla options -valikosta Open. Tämä käynnistää *EMMS_applicationCmdOpen* -komennon, jonka käsittely tapahtuu käyttöliittymäluokassa. *ReadmessagesL* -funktiolla luetaan jälleen viestin tiedot taulukkoon. *CMMS_applicationContainer* -luokan funktio *DisplayDescriptionL* tutkii, sisältääkö taulukko dataa. Jos paluuarvo on tosi, tyhjennetään näyttö ja näytetään näytöllä viestin tekstisisältö.



Kuvio 15. Sekvenssikaavio viestin avaamisesta

Kuviossa 16 on mallinnettu tilanne viestin poistamisesta. Käyttäjä valitsee yksitellen inboxissa poistettavan viestin ja options -valikosta Remove. *EMMS_applicationCmdRemove* -komento käsitellään käyttöliittymäluokassa. Viesti poistetaan *DeleteMessageL* -funktion avulla globaalista inbox -kansioista. *CompleteReadL* -funktio palauttaa poistettavan viestin yksilöivän id:n. Varsinainen viestin poistaminen tapahtuu sovellusmoottorin *DeleteMessagesL* -funktiossa. Tämän jälkeen *ReadMessagesL* -funktioilla luetaan inboxin viestit uudestaan ja päivitetään lista piirtämällä se uudestaan.



Kuvio 16. Sekvenssikaavio viestin poistamisesta

7.3 Toteutus

Vaatimusmäärittelyn mukaan yksi sovelluksen toiminnallisista tavoitteista oli viestin automaattinen noutaminen MMSC palvelimelta sen jälkeen, kun päätelaitteeseen on saapunut ilmoitus uudesta viestistä. Tämä oli tarkoitus toteuttaa kuviossa 6 esitetyn mallin mukaisesti, mutta Symbian OS käyttöjärjestelmä sisältää jo tämän toiminnallisuuden. Järjestelmä luo GPRS-yhteyden automaattisesti ja noutaa viestit MMSC viestipalvelimelta. Viestin saapumisesta ilmoitetaan käyttäjälle äänimerkillä ja ilmoitusviesti näytetään näytöllä. Tässä opinnäytetyöprojektissä toteutettu MMS sovellus sisältää myös toiminnallisuuden ilmoitustekstin näyttämiseen.

7.3.1 Lähtökohta

MMS sovellus toteutettiin luomalla ensin MS Visual Studion Application Wizard –ohjelmalla projektipohja. Series 60 SDK:n mukana tuli asennuspaketti projektipohjan määrittelemiseksi AppWizard ohjelmaan. AppWizard loi valmiiksi valitun sovellusarkkitehtuurin mukaiset perusluokat ja määrittelytiedostot. Projektiin liitettiin mukaan MMSDemo esimerkisovelluksen sovellusmoottori (engine). Varsinainen sovelluskehitys tehtiin Visual Studio C++ ohjelmalla, johon liitettiin SDK:n emulaattori testaamista ja debuggaamista varten.

7.3.2 Omat lisäykset

Aluksi sovelluksen valikkorakenteeseen tehtiin muutos. Sovelluksen resurssitiedostoon MMS_application.rss määriteltiin options –valikon vaihtoehdoiksi Open, Remove, Open inbox ja Exit. Lisäksi jokaiselle valikon vaihtoehdolle määriteltiin omat komennot MMS_application.hrh tiedostoon. Sovelluksen käyttöliittymäluokka käsittelee valikon komennot *HandleCommandL()* –funktiossa.

Toteuttamalla käyttöliittymäluokassa funktio *DynInitMenuPaneL* voidaan dynaamisesti muokata options –valikon vaihtoehtoja. Kun sovellus käynnistetään, valikossa on näkyvillä Open inbox ja Exit. Kun inbox on avattu eli lista viesteistä on näkyvillä näytöllä, valikon vaihtoehdoiksi muuttuu Open, Remove ja Exit.

Sovelluksen käyttöliittymäluokan konstruktorissa heti sovellusmoottorin alustamisen jälkeen sovellus tutkii, onko inboxissa uusia lukemattomia viestejä. Jos sovellusmoottorin funktio *NewMessages()* palauttaa arvon, joka on suurempi kuin nolla, näytetään päätelaitteen näytöllä teksti ”Multimedia message received”. Tämä toteutettiin CMMS_applicationContainer –luokkaan toteutetun *DisplayTheTextL()* –funktion avulla. ”//EB” –etuliitteellä olevien viestien poimiminen toteutettiin *DisplayTheListboxL* –funktioon. Sovellus etsii viestin sisällöstä ”//EB” –merkkijonoa ja palauttaa arvon nolla, jos merkkijono löytyy viestin alusta. Tämän jälkeen sisältö liitetään listboxiin.

Viestin avaamisen toteuttamiseksi täytyi muokata hieman käyttöliittymäluokkaan MMSDemo esimerkistä liitettyä *ReadMessagesL()* funktiota. Funktio määriteltiin vastaanottamaan toinen parametri (aParam), jonka avulla tunnistetaan, ollaanko avaamassa inboxia (aParam=1) vai valittua viestiä listalta (aParam=2). Molemmissa tapauksissa viestit täytyy ensin lukea taulukkoon, joka lähetetään parametrina *DisplayTheListboxL()* tai *DisplayDescription()* –fuktiolle. Jälkimmäinen funktio tyhjentää näytön ja näyttää viestin tekstisisällön näytöllä.

Viestin poistaminen oli toteutettu valmiiksi MMSDemo sovellusmoottorissa. Options –valikon Remove –vaihtoehdolle määritelty komento täytyi vain muuttaa *HandleCommandL()* –funktioon.

7.3.3 Ongelmatilanteet

MMS sovelluksen toteutusvaiheessa ilmeni ongelmia viestissä liitetiedostona saapuvien kuva- ja äänitiedostojen avaamisen kanssa, minkä vuoksi projektin loppuvaiheessa näiden ominaisuuksien toteuttamisesta päätettiin luopua. Liitetiedostojen avaamiseen ja niiden sisältämän datan esittämiseen käyttäjälle on useita vaihtoehtoja. Yksi vaihtoehto olisi ollut käyttää Series 60 ohjelmistoalustan sisältämää valmista, yksinkertaista MMSViewer ohjelmaa.

Toinen vaihtoehto on tehdä oma toteutus tiedostojen käsittelyyn. Kuvien käsittely olisi vaatinut oman erillisen sovellusmoottorin, jossa käytetään MMF multimediatekehyksessä olevaa ICL (Image Converter Library) –rajapintaa. Kuvat muunnetaan ensin bitmap muotoon, jonka jälkeen niitä voidaan piirtää näytölle.

Lisäksi Series 60 ohjelmistoalusta sisältää erilaisia sovelluksia tietynlaisen datan esittämistä varten. Näitä sovelluksia voidaan avata ja käyttää sulautetusti muiden sovellusten sisällä, jos isäntäsovellus on määritelty kykeneväksi tähän. Sovelluksen AIF (Application Information File) tiedostossa määritellään sovelluksen tukemat MIME tyypit. Esimerkiksi jpeg –tyyppisten kuvien MIME tyyppi on image/jpeg. Tämän perusteella ohjelmistoalusta tunnistaa käsiteltävän datan tyyppin ja osaa käynnistää oikeanlaisen sovelluksen datan esittämiseen. Kuvatiedostojen esittämiseen käytetään Photoalbum sovellusta.

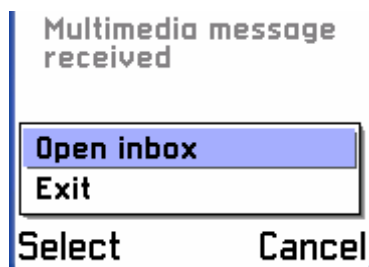
7.4 Testaus

MMS sovelluksen testaamista suoritettiin kolmessa eri vaiheessa. Järjestelmätestauksesta tehtiin erillinen testaussuunnitelma ja laadittiin testitapaukset. Testaamisessa käytettiin työkaluina SDK:n emulaattoria ja kohdepuhelinia.

Forum Nokian internet sivustolta on saatavissa myös PC-pohjainen ohjelma, jolla voidaan luoda ja lähettää multimediatestejä emulaattoriin. Ohjelman mukana tuli karsittu Series 60 emulaattori, johon ei pystynyt asentamaan MMS sovellusta, joten tätä testaustyökalua ei voitu käyttää tässä projektissa.

Yksikkötestaus

Sovellukseen itse toteutetut moduulit testattiin yksikkötestauksen periaatteiden mukaisesti. Options –valikon toimiminen ja tekstin kirjoittaminen näytölle testattiin ensin emulaattorilla, jotta esimerkiksi tekstin layout saatiin halutun näköiseksi. Kuviossa 17 on emulaattorin näyttö tilanteessa, jossa sovellus on käynnistetty ja options valikko on avattu.



Kuvio 17. MMS sovelluksen aloitusvalikko ja ilmoitusviesti uusista viesteistä inboxissa.

Ehtolauseita, joiden tarkoituksena on tarkistaa onko inboxissa uusia lukemattomia viestejä, muutettiin siten, että ilmoitusteksti näkyi näytöllä aina, kun sovellus avattiin. Yksikkötestausvaiheen jälkeen muita testausvaiheita varten ehtolauseita muutettiin vastaamaan todellista tilannetta. Viestien poimiminen viestivirrasta sekä viestin avaaminen ja poistaminen voitiin testata vasta integrointi- ja järjestelmätestausvaiheissa, koska emulaattorilla ei voitu simuloida multimediatestien vastaanottamista.

Integroititestaustaus

Integroititestaamista tehtiin siinä vaiheessa, kun MMSDemo esimerkin sovellusmoottori liitettiin mukaan projektiin ja sovelluksen muihin luokkiin liitettiin uusia funktioita. Integroititestaustuksessa käytettiin kokoavaa (top-down) integrointistrategiaa. Erillisinä testitapauksina testattiin inboxin avaamiseen, viestin avaamiseen ja viestin poistamiseen osallistuvat moduulit.

Sovelluksen toiminta tilanteessa, jossa inbox on tyhjä, voitiin testata emulaattorissa, mutta muut integroititestaustuksen vaiheet täytyi tehdä oikeassa päätelaitteessa. Tätä varten puhelimeen täytyi ensin lähettää ainakin yksi multimediatesti, jotta lista inboxissa olevista viesteistä voitiin näyttää. ”//EB” etuliitteellä saapuneiden viestien poimiminen todennettiin lähettämällä testipuhelimeen myös sellainen viesti, jossa ei ollut mukana etuliitettä.

MMS sovellus asennetaan matkapuhelimeen .sis asennuspakettina, jolloin .pkg tiedosto täytyy kääntää komennolla ”makesis tiedostonimi.pkg”. Pkg-tiedosto määrittelee, kuinka asennustiedosto tulee kääntää. Asennuspaketti voidaan siirtää puhelimeen esimerkiksi infrapunaa tai bluetooth yhteyden avulla. Asennustiedosto saapuu puhelimeen tekstiviestinä ja käynnistää asennuksen automaattisesti. Jos laitteessa on aikaisempi versio samasta sovelluksesta, asennusohjelma kysyy, korvataanko aikaisempi versio.

Integroititestaustavaiheessa ilmeni ongelmia *CMMS_applicationContainer* luokkaan toteutetun *DisplayDescriptionL()* ja *CMMS_applicationAppUi* luokan *ReadMessagesL()* funktioiden toiminnassa. Samoin sovelluksen alustamisen yhteydessä toteutetussa saapuneiden uusien viestien tarkistamisessa ilmeni ongelmia. Moduulien toimintaa ei saatu korjattua projektin aikataulun puitteissa.

Järjestelmätestaustaus

Sovelluksen järjestelmätestaustavaiheessa testattiin MMS sovelluksen toiminnallisuutta kokonaisuutena. Järjestelmätestaustuksesta tehtiin erillinen testisuunnitelma vaatimusmäärittelyssä määriteltujen tavoitteiden pohjalta. Järjestelmätestaustavaiheessa ajettiin seuraavat suunnitelman mukaiset testitapaukset:

- sovelluksen käynnistäminen
- viestien tutkiminen ja poimiminen viestivirrasta
- viestilistan luominen
- ilmoituksen näyttäminen
- viestin avaaminen
- viestin poistaminen.

Testitapauksista kaikki muut paitsi ilmoituksen näyttäminen ja viestien avaaminen läpäisivät järjestelmätestauksen hyväksytysti. Uuden viestin saavuttua puhelimeen sovellus ei näyttänyt ”Multimedia message received” –tekstiä näytöllä. Viestin avaaminen ei läpäissyt järjestelmätestausta, koska viestin tekstisisältö ei näkynyt näytöllä.

7.5 Kehitysehdotukset

Opinnäytetyön tuloksena syntynyttä MMS sovellusta voi kehittää edelleen toteuttamalla kaikenlaisten multimediamiestien käsittelyn ja liitetiedostojen avaamisen. Sovelluksen arkkitehtuuri on myös muutettavissa esimerkiksi view-arkkitehtuurin mukaiseksi. Käyttöliittymäkomponentteja voidaan vaihdella tarpeen mukaan, esimerkiksi inboxissa olevat viestit voidaan esittää erilaisessa listakontrollissa.

Symbian OS käyttöjärjestelmä ja Series 60 ohjelmistoalusta yhdessä tarjoavat erinomaisen alustan kehittää omia sovelluksia älypuhelimiin. Internetistä on saatavilla paljon erilaisia esimerkkisovelluksia, joiden pohjalta voi kehittää omia sovelluksia erilaisiin käyttötarkoituksiin.

8 POHDINTA

Tämän opinnäytetyön tekeminen oli haasteellinen tehtävä, koska aihealue ei ollut ennestään tuttu eikä mobiiliympäristöihin ja -ohjelmointiin oltu tutustuttu myöskään koulussa. Harjoittelupaikkani ansiosta kiinnostuin kuitenkin aiheesta ja erityisesti mahdollisuus kehittää omia sovelluksia älypuhelimiin motivoi opiskelemaan uutta asiaa. Symbian ohjelmointi on paljon C++:n kaltaista ja olio-ohjelmointipohjaista, joten sen perusteella aihe ei tuntunut ylivoimaisen vaikealta.

Symbianista on olemassa vielä melko vähän kirjallista aineistoa ja valtaosa siitä on englanninkielistä. Sen sijaan Series 60 ohjelmistoalustaa käyttäville ohjelmistokehittäjille on tarjolla runsaasti hyvää materiaalia Forum Nokian internet sivustolla, mutta sekin englanniksi. Kieli ei kuitenkaan tuottanut ongelmia opinnäytetyön tekemisessä. Ohjelmistotestaamisesta sen sijaan oli tarjolla paljon suomenkielistä aineistoa.

Opinnäytetyön teoriaosa esittelee Symbian OS käyttöjärjestelmän rakennetta ja toiminnallisuutta, mutta niillä tiedoilla ei ollut kovin paljon käyttöä MMS sovelluksen ohjelmoimisessa. Teoriaosassa esiteltiin Symbian OS 7.0s version MMF multimediakehys, jonka rakenteeseen tutustuin tarkemmin kuvatiedostojen käsittelyn yhteydessä. Käyttöjärjestelmän esittelyssä pysyttiin hyvin yleisellä tasolla, koska yksityiskohtaisilla teknisillä tiedoilla järjestelmän ominaisuuksista ei olisi ollut käyttöä. Tässä opinnäytetyössä keskityttiin enemmän tarkastelemaan käyttöliittymätason ohjelmistosuunnittelua.

Series 60 ohjelmistoalustan teoria antoi hyvän yleiskuvan käyttöliittymästä ja siitä, millä tavalla Series 60 alustalle ohjelmoidaan sovelluksia. Sovelluksen rakenteesta sai hyvän

käsityksen jo ennen ohjelmoinnin aloittamista. Teoriaosassa esitetty malli pätee täysin yhtenevästi käytännön osassa toteutettuun MMS sovellukseen.

MMS sovelluksen testausvaiheessa toteutettiin teoriaosassa esitetyt testausvaiheet, mutta vain järjestelmätestauksesta tehtiin erillinen suunnitelma. Yksikkötestausta ei voitu suorittaa kaikille moduuleille, koska emulaattorissa ei voitu toteuttaa multimediaviestien vastaanottamista. Koska emulaattoria voitiin käyttää melko vähän MMS sovelluksen testaamisessa, se vaikeutti virheiden paikantamista epäselvissä tapauksissa.

Opinnäytetyön tekeminen eteni projektiluontoisesti ja teoriaosuuden kirjoittamisen jälkeen ryhdyin opiskelemaan FCS Partners:n kurssimateriaalin pohjalta Symbian ohjelmointia Series 60 ohjelmistoalustalle. Materiaali sisälsi Power Point kalvoja ja laboratorio harjoitustöitä. Aineisto oli erittäin selkeää ja sen pohjalta oli helppo opiskella uutta asiaa. Pelkän internetistä ja kirjoista saatavan materiaalin pohjalta opinnäytetyön tekeminen olisi ollut paljon työläämpää.

Varsinainen oma ohjelmointityö eteni pitkälti ”kokeile ja koodaa” –periaatteen mukaisesti, koska Series 60 ohjelmistoalusta ei ollut ennestään tuttu ja sovellusarkkitehtuurin sisäistämiseen kului melko paljon aikaa. Jouduin useita kertoja luomaan koko projektin uudestaan, kun sovellus meni täysin jumiin eikä käynnistynyt ollenkaan.

Kuva- ja äänitiedostojen käsittelyn toteuttaminen oli ongelmallista, koska näistä ei ollut kovin paljon esimerkkejä tarjolla ja itsellä ei vielä ollut tietoa siitä, millaisia luokkia ja funktioita olisi ollut parasta käyttää. Tämän ongelman kanssa aikaa kului liiankin paljon, mutta projektin lähestyessä loppuaan kuvien käsittelyn toteuttamisesta päätettiin luopua. Sovelluksen muiden toimintojen toteuttaminen kävi nopeammin, mutta myös niiden ohjelmoimisessa oli pieniä ongelmia. Usein metodit jonkin toiminnon toteuttamiseksi olivat oikeat, mutta tiedon soveltaminen käytäntöön oli vielä puutteellista. Series 60 SDK:n mukana tulleesta helpistä oli ohjelmointityössä suurta apua.

Opinnäytetyöprojektin aikataulu oli laadittu melko väljäksi ja projekti eteni hyvin sen mukaan. Siitä huolimatta lopussa tuli kuitenkin kiire, koska kuvienkäsittelyongelman kanssa kului niin paljon aikaa. Projektin tavoitteita ei oltu priorisoitu tarkemmin vaatimusmäärittelyssä, vaan kaikille vaatimuksille oli asetettu prioriteetiksi ’korkea’. Tämä aiheutti hieman sekaannusta, koska toteutettavien ominaisuuksien priorisointi selkiytyi vasta

projektin lopussa. Omasta mielestäni viestien avaaminen ja liitetiedostojen näyttäminen olisi ollut yksi tärkeimmistä tavoitteista. Lisäksi MMS sovellusta ei voitu täysin toteuttaa vaatimusmäärittelyn mukaisesti mm. projektin aikana ilmenneiden ohjelmointiongelmien takia.

Lopputuloksena MMS sovelluksesta tuli hyvin pelkistetty. Mielestäni kuitenkin tässä työssä käytännön osassa tehtyä sovellusta suurempi merkitys on sillä, että opin Symbian ohjelmoinnin perusteet ja pystyn jatkossa kehittämään omia sovelluksia älypuhelimiin. Se on todella kiinnostava ja innostava ominaisuus uuden teknologian matkapuhelimissa. Mahdollisesti jatkan MMS sovelluksen kehitystyötä vielä tämän projektin jälkeen, koska haluan nähdä millainen sovelluksesta olisi tullut jos se olisi toiminut oikein.

LÄHTEET

Julkaistu materiaali:

Arokoski, Jääskeläinen, Kontio, Köykkä, Raatikainen, Tervo & Vierimaa. 2002.

Mobiiliteknologiat. Helsinki: Edita Prima Oy.

Haikala, I. & Märijärvi, J. 2002. Ohjelmistotuotanto. Helsinki. Satku.

Helsingin Teknillinen Korkeakoulu. 2002. Mobiili Internet. Saatavilla WWW-muodossa

<http://keskus.hut.fi/opetus/s38119/k02/raportti.pdf> (Luettu 13.2.2004)

Jipping, M. 2002. Symbian OS Communications Programming. West Sussex. John Wiley & Sons, Ltd.

Nokia 2004 a. Overview of Symbian OS 7.0s. Saatavilla WWW-muodossa

http://www.forum.nokia.com/main/0,6566,st_e_22,00.html (Luettu 19.2.2004)

Nokia 2004 b. Mikä on SyncML? <http://www.nokia.fi/puhelimet/teknologiat/syncml/>

(Luettu 22.2.2004)

Nokia 2004 c. Useimmin kysytyt kysymykset.

<http://www.nokia.fi/puhelimet/teknologiat/mms/faq.html> (Luettu 12.3.2004)

Nokia 2004 d. Kopioitu dokumentti internetistä: Designing Applications for Smartphones – Series 60 Platform Overview.

http://www.forum.nokia.com/main/1,6566,21,00.html?fsrParam=1-3-/main/1,,21_10,00.html&fileID=2184 (Luettu 13.3.2004)

Nokia 2004 e. Kopioitu dokumentti internetistä: Developer Platform 2.0 for Series 40: Introductory White Paper.

<http://www.forum.nokia.com/main/1,,040,00.html?fsrParam=3-3-/main.html&fileID=3772> (Luettu 14.3.2004)

Nokia 2004 f. Tiedote Nokian vuoden 2003 viimeisen neljänneksen tuloksesta.

<http://www.nokia.com/2003/4Q/files/4Q-PR-F.pdf> (Luettu 14.3.2004)

Nokia 2004 g. Press release. http://press.nokia.com/PR/200310/922404_5.html (Luettu 14.3.2003)

Nokia 2004 h. Kopioitu dokumentti internetistä: Developer Platform 2.0 for Series 90: Introductory White Paper.

<http://www.forum.nokia.com/main/1,,040,00.html?fsrParam=3-3-/main.html&fileID=4035> (Luettu 14.3.2004)

Nokia 2004 i. Kopioitu dokumentti internetistä: How to Create MMS Services.

www.forum.nokia.com/main/1,6566,1_2_7_30,00.html. (Luettu 19.4.2004)

Nokia 2004 j. Kopioitu MMS demo esimerkisovellus internetistä.

<http://www.forum.nokia.com/main/1,,040,00.html?fsrParam=2-3-/main.html&fileID=2911> (Poimittu 1.4.2004)

Sanoma Magazines Finland 2003. Saatavilla WWW-muodossa

<http://www.itviikko.fi/uutiset/uutinen.asp?UutisID=59331> 9.2.2004 (Luettu 14.2.2004)

Satukangas, A. 2003. Pro gradu -tutkielma, Yksikkö ja integrointitestauksen menetelmät ja mittarit. Saatavilla WWW-muodossa

<http://ethesis.helsinki.fi/julkaisut/mat/tieto/pg/satukangas/yksikkoj.pdf> (Luettu 12.3.2004)

Symbian DevNet 2002. Colin Turfus. Using MMS on Symbian OS phones. Saatavilla WWW-muodossa

<http://www.symbian.com/developer/techlib/papers/mms/mms.pdf> (Luettu 28.8.2004)

Symbian Ltd 2004 a. <http://www.symbian.com/about/ownership.html> (Kuvio poimittu 14.2.2004)

Symbian Ltd 2004 b. Symbian's vision. Saatavilla WWW-muodossa

<http://www.symbian.com/about/vision.html> (Luettu 14.2.2004)

Symbian Ltd 2004 c. Why is a different Operating System needed? Saatavilla WWW-muodossa <http://www.symbian.com/technology/why-diff-os.html> (luettu 14.2.2004)

Symbian Ltd 2004 d. Symbian OS Version 7.0s Functional description. Saatavilla WWW-muodossa http://www.symbian.com/technology/7.0s_functional_description2.1.pdf kesäkuu 2003 (Luettu 17.2.2004)

Tampereen Teknillinen yliopisto 2003 a. MMS perusteet.. Saatavilla WWW-muodossa
http://www.ac.tut.fi/aci/courses/7601010/2003_audi/pdf/MMS_perusteet.pdf
(Luettu 19.2.2004)

Tampereen Teknillinen yliopisto 2003 b. Langattomien sovellusten testaus & niiden rooli
osana kokonaisuutta. Saatavilla WWW-muodossa
http://www.ac.tut.fi/aci/courses/7601010/2003_audi/pdf/Ch14,_Ch15.pdf kevät
2003 (Luettu 12.3.2004)

Telefonaktiebolaget L M Ericsson 1994 – 2003. WCDMA. Saatavilla WWW-muodossa
<http://www.ericsson.com/fi/technology/WCDMA.shtml> 21.5.2003. (Luettu
19.2.2004)

Julkaisematon materiaali:

FCS Partners: Series 60 C++ (NOC) kurssimateriaali (Nokia 2003)