

-Symbian-

Profesor: Stancescu Stefan

Masterand: Stanciu Valerian

Program: IISC

An I

Cuprins

Capitolul 1: Introducere	3
Capitolul 1.1: Domenii tehnologice si pachete	3
Capitolul 1.2: Kernelul Symbian.....	4
Capitolul 1.3: Design.....	4
Capitolul 2: Platforme Symbian.....	5
Capitolul 2.1: Platforma S60	6
Capitolul 2.2: Platforma S40	6
Capitolul 3: Interconectare cu Java.....	7
3.1: Java Micro Edition	8
3.1.1. Configuratii	9
3.1.2. Profiluri	10
3.1.3. Pachete optionale	11
3.2 Java ME vs. Symbian C++.....	11
Capitolul 4. Comparatie cu alte sisteme de operare mobile.....	13
3.1 Symbian vs. Android.....	14
3.2 Symbian vs. Windows Mobile	15
Capitolul 5. Dezvoltarea unei aplicatii pentru Symbian	16
5.1 Echipamente de dezvoltare.....	17
5.2. Aspecte de luat in seama in dezvoltarea unei aplicatii.....	17
Procesorul	17
Memoria.....	18
Lungimea de banda pentru comunicatii	18
Securitatea.....	5
Capitolul 6. Bibliografie	19

Capitolul 1: Introducere

Dezvoltatorii de aplicații mobile sunt în continua dezbatere în legatura cu platforma dezvoltare optima în ceea ce privește aplicațiile mobile. Răspunsul la această întrebare este simplu: nu există nicio platformă unică să poată fi considerată cea mai bună de pe piață.

Symbian este un sistem de operare mobil (OS) și platforma de calcul concepute pentru smartphone-uri și în prezent, menținut de către Accenture. Platforma Symbian este succesorul Symbian OS și Nokia Series 60. Cea mai recentă versiune, Symbian ^ 3, a fost lansat oficial în la sfârșitul anului 2010, folosit pentru prima dată la Nokia N8. În luna mai 2011 un update, Symbian Anna, a fost anunțat oficial, urmat de Symbian Belle în august 2011.

Platforma Symbian a fost creată prin unirea și integrarea atributelor software, la care au contribuit Nokia, NTT DoCoMo, Sony Ericsson și Ltd. Symbian, având la baza caracteristicile Symbian OS, platforma S60, și părți ale interfețelor de utilizator UIQ și MOAP.

La 11 februarie 2011, Nokia a anunțat un parteneriat cu Microsoft, prin care adopta Windows Phone 7 pentru smartphone-uri, reducând astfel numărului de dispozitive care rulează Symbian. Ca o consecință, utilizarea platformei Symbian pentru aplicațiile mobile a scăzut rapid. Un studiu efectuat în iunie 2011 a indicat că peste 39% din dezvoltatorii mobili care foloseau atunci Symbian, aveau planificat să renunțe la platforma.

La 5 aprilie 2011, Nokia a încetat să mai ofere orice parte a software-ului Symbian open source și a redus colaborarea sa la un grup mic de parteneri din Japonia.

Capitolul 1.1: Domenii tehnologice și pachete

Designul Symbian este împărțit în domenii tehnologice, fiecare dintre ele cuprinzând un număr de pachete de programe. Fiecare domeniu tehnologic are propria foaie de parcurs, și Symbian Foundation are o echipă de manageri de tehnologie care gestionează aceste foi de parcurs ale domeniilor tehnologice.

Fiecare pachet este alocat exact unui domeniu tehnologic, bazat pe zonă funcțională generală la care contribuie și prin care poate fi influențat. Prin gruparea pachetelor conexe prin teme, Symbian Foundation speră să încurajeze formarea unei comunități puternice în jurul lor și pentru a genera discuții și dezbateri.

Pachetele sunt deținute și întreținute de către un proprietar de pachet, un individ dintr-o de organizație membră a fundației Symbian, care acceptă contribuțiile de cod din comunitatea Symbian și este responsabilă pentru pachet.

Capitolul 1.2: Kernelul Symbian

Kernel-ul Symbian (EKA2) suporta răspuns suficient de rapid în timp real pentru a construi un telefon single-core în jurul acestei calități, adică, un telefon în care un singur nucleu de procesor execută atât cererile de utilizator și stiva de semnalizare. Nucleul de timp real are o arhitectură microkernel conținând doar primitivele minime, cel mai de bază și funcționalitate pentru robustețe maximă, disponibilitate și capacitate de reacție. Acesta a fost numit un nanokernel, deoarece are nevoie de un nucleu extins pentru a pune în aplicare orice alte abstracțiuni. Acesta conține un sistem de gestiune a timpului, un sistem de gestionare a memoriei și a aparatului de utilizare, cu servicii de rețea, telefonie și sistemul de fișiere de sprijin în stratul OS Servicii (OS Service Layer) sau stratul de servicii de bază (Base Services Layer). Includerea de drivere de dispozitiv reprezintă kernel-ul nu este un microkernel adevărat.

Kernelul Symbian dispune de următoarele facilitati:

- Multi-tasking: planificarea evenimentelor astfel incat CPU-ul sa poata rula mai multe task-uri simultan (time-sharing, multiprogramming)
- RTOS (real-time operationg system)
- Pre-emptive: intreruprerea unui task fara acordul acestuia, cu intentia de a reveni mai tarziu la acel task
- XIP (eXecute In Place): executarea programelor direct de pe dispozitivul de stocare, fara transfer in RAM

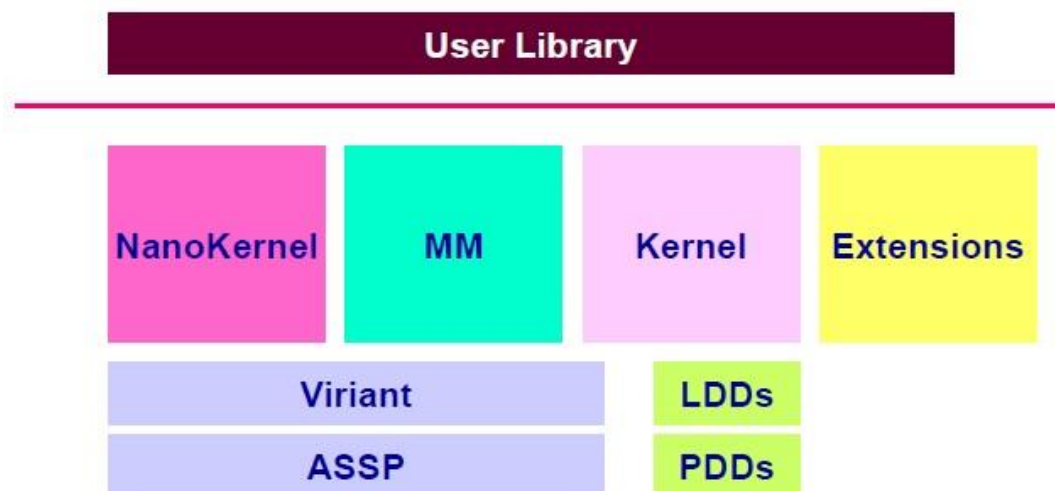


Figura 1. Arhitectura unui kernel Symbian

Capitolul 1.3: Design

Symbian dispune de caracteristicile pre-emptive multitasking și protecția memoriei, la fel ca alte sisteme de operare (în special cele create pentru a fi utilizate pe computere desktop). Abordarea EPOC a multitasking-a fost inspirat de VMS și se bazează pe evenimente bazate pe server asincron.

Symbian OS a fost creat având în minte trei principii de proiectare a sistemelor:

- integritatea și securitatea datelor de utilizator este primordială
- timpii de utilizare nu trebuie irosiți
- toate resursele sunt limitate

Securitatea

Securitatea este un factor important. Aplicațiile native au acces nelimitat la dispozitiv. O aplicație nativă poate de exemplu să acceseze direct interfața rețelei dispozitivului, memoria de stocare, interfața telefonului, mesageria, etc. Acest lucru oferă multe posibilități pentru dezvoltator, dar aceste posibilități pot de asemenea să fie folosite în mod necorespunzător. Din păcate, nu există niciun mod de a determina dacă un anumit software este sau nu sigur. Este posibil să folosim semnături și certificate pentru a oferi utilizatorului un anumit nivel de încredere că aplicația pe care se instalează are o sursă de încredere. Totuși, aceste măsuri nu fac aplicația sigură, ele doar asigură că aplicația este de la provider-ul respectiv. În ciuda acestor lucruri, tehnologii de comunicații de date standard și sigure sunt bine suportate de Symbian OS. Aceste tehnologii dau dezvoltatorilor instrumente pentru a crea aplicații de comunicații sigure. [5]

Pentru a urmări cel mai bine aceste principii, Symbian folosește un microkernel, are o abordare cerere-și-callback la servicii, și menține separarea între interfața de utilizator și a motor. Sistemul de operare este optimizat pentru dispozitive cu consum redus de energie pe baza de baterie și pentru sisteme bazate pe ROM (de exemplu, caracteristici cum ar fi XIP și re-entrancy în bibliotecile partajate). Cererile, precum și sistemul de operare în sine, urmăresc un design orientat-obiect: Model-View-Controller (MVC).

Există un accent puternic pe conservarea resurselor, care este exemplificat prin expresii de programare Symbian, specifice cum ar fi descriptorii și o stivă de curățare (cleanup stack). Există metode similare pentru a conserva spațiul de pe disc, deși discuri de pe dispozitivele Symbian sunt, de obicei, de memorie Flash. Mai mult, toată programarea Symbian este bazată pe evenimente (event based), iar unitatea centrală de prelucrare (CPU) este comutată într-un modul de consum redus atunci când cererile nu au direct de-a face cu un eveniment. Acest lucru se face printr-un idiom de programare numit obiecte active. În mod similar, abordarea Symbian în ceea ce privește thread-urile și procesele este să reducă costurile generale.

Capitolul 2: Platforme Symbian

Capitolul 2.1: Platforma S60

Platforma S60 (fostă Series 60 User Interface) este o platformă software pentru telefoanele mobile care rulează pe sistemul de operare Symbian. A fost creat de Nokia, care a distribuit-o open source și a contribuit și la Symbian Foundation. S60 a fost folosit de producătorii de dispozitive mobile, inclusiv Siemens mobile, Lenovo, LG Electronics, Panasonic și Samsung. Sony a creat cu software-ul împreună cu Nokia. Symbian este sistemul de operare cel mai popular în rândul smartphone-uri de pe piață, cu 37,6% din vânzările totale ale sectorului, cu 111.6 milioane de telefoane mobile vândute în anul 2010.

În plus față de producători, comunitatea include:

- Companii de integrare software cum ar fi Sasken, Elektrobit, Teleca, Digia, Mobica, Atelier.tm
- Companiile de semiconductori Texas Instruments, STMicroelectronics, Broadcom, Sony, Freescale Semiconductor, Samsung Electronics
- Operatori, cum ar fi Vodafone și Orange, care dezvoltă și furnizează S60 bazate pe aplicații și servicii mobile
- Dezvoltatorii de software și furnizori independenți de software

S60 este format dintr-o suită de biblioteci și aplicații standard, cum ar fi telefonie, manager de informații cu caracter personal (PIM) și de playere multimedia pe bază de Helix. Acesta este destinat să ruleze pe telefoane moderne cu ecrane color de mari dimensiuni, care sunt de obicei cunoscute sub numele de smartphone-uri.

Software-ul S60 este un standard mai mulți furnizori pentru smartphone-uri care sprijină dezvoltarea de aplicații în Java MIDP, C + +, Python, și Adobe Flash. Inițial, caracteristica cea mai distinctivă a telefoanelor S60 a fost faptul că au permis utilizatorilor să instaleze aplicații noi după cumpărare. Spre deosebire de o platforma desktop standard, cu toate acestea, aplicații built-in sunt rareori modernizate de către vânzător, în afara de repara bug-urile. Noile caracteristici sunt adăugate doar în timp ce acestea sunt în curs de dezvoltare, mai degrabă decât după eliberarea publică. Anumite butoane sunt standardizate, cum ar fi o tastă de meniu, joystick cu patru direcții (d-pad), tastele stânga și dreapta moale și o tastă de ștergere. [7]

Capitolul 2.2: Platforma S40

Seria 40 este o platformă software și interfața cu utilizatorul (UI) pentru aplicații, pe o gamă largă de telefoane Nokia cu facilități mid-tier, precum și pentru linia de telefoane de lux Vertu. Este cea mai utilizată platforma pentru telefonul mobil din lume pe scară largă și e găsit în sute de milioane de dispozitive. În 2010 erau aproximativ 1.5 miliarde de dispozitive.

Acesta oferă aplicații de comunicare cum ar fi telefonie prin IP (VoIP), mesagerie, client de e-mail cu capacitățile POP3 și IMAP4 și browser Web, aplicații media cum ar fi aparat de fotografiat, înregistrare video, player audio/video și radio FM, precum și agenda telefonică și alte management de informații personale (PIM), aplicații cum ar fi calendar și sarcini. Gestionarea

fișierelor de bază, ca în Seria 60, este prevăzută în aplicații și dosarele Galerie și subfoldere. Aplicațiile instalate de utilizator pe Seria 40 sunt, în general, aplicații mobile Java. Aplicații Flash Lite sunt de asemenea posibile, dar mai ales pentru screensaver-e.

Browser-ul web integrat poate accesa mare parte din conținutul web prin intermediul v furnizorului de servicii XHTML/HTML. Cea mai recentă versiune a Seriei 40, numita Seria 40 Ediția a 6-a a introdus un browser nou, bazat pe componentele open source WebKit (WebCore și JavaScriptCore). Noul browser ofera suport pentru HTML 4.01, CSS2, JavaScript 1.5 și Ajax. De asemenea, la fel ca seria high-end S60, Seria 40 poate rula browser-ului Opera Mini pentru a îmbunătăți experiența utilizatorului cu navigarea pe Web.

Seria 40 este o platformă software embedded, care este deschisă pentru dezvoltarea de software prin intermediul tehnologii de dezvoltare de aplicații și de conținut standard sau de facto. Acesta susține Java MIDlet-uri, și anume Java MIDP CLDC, care oferă capabilități localizare, comunicare, mesaje, mass-media, și grafica. S40 suportă, de asemenea, aplicații Flash Lite.

Seria 40 este un sistem de operare mai simplu decât S60, care este mai high-end (care se bazează pe multi-taskingul sistemului de operare Symbian). Deoarece dispozitivele S40 nu acceptă cu adevărat multi-tasking și nu au un cod nativ API pentru terți, interfața de utilizare al acestuia poate părea să fie mai receptivă și mai rapidă decât alte platforme Nokia pe hardware similar. [8]

Capitolul 3: Interconectare cu Java

Java Runtime pentru Symbian este compatibil cu Java ME, mediul Java Runtime conceput pentru dispozitivele Nokia Symbian. Pornind de la S60, a 5-a ediție de Java Runtime pentru Symbian va avea versiuni separate din platforma S60, cu scopul de a oferi mai des noi versiuni Java pe piață. Factori cheie pentru în acest sens sunt, de exemplu, timpul mai scurt de lansare pentru noi caracteristici și o posibilitate de a actualiza Java Runtime pentru Symbian în mod independent prin fișiere SIS descărcabile, și, de asemenea, în viitor, prin intermediul Nokia SW Update.

Nokia a întrerupt utilizarea denumirii produsului "S60" și "Symbian" este numele pentru viitor. Primul Java Runtime cu această schimbare este Java Runtime 2.1 pentru Symbian. Versiunile mai vechi (JRT 1.3 și JRT 1.4) sunt numite Java Runtime 1.3 pentru S60 și Java Runtime 1.4 pentru S60. Numele cel mai des întâlnite sunt Java Runtime pentru S60 și Java Runtime pentru Symbian.

Java are o lungă istorie pe sistemul de operare Symbian care datează de la Symbian OS Versiunea 5 (lansat în 1999). Acest Java inițial s-a bazat pe platforma Sun JDK 1.1.4. Pentru versiunea majoră următoare, Symbian a decis să profite de amprenta de memorie redusă oferită de Personal Java și a utilizat caietul de sarcini PersonalJava 1.1.1 ca bază pentru punerea în aplicare Java. Această versiune, sistemul de operare Symbian versiunea 6.0, a devenit disponibil în 2000.

Personal Java a fost precursorul a J2ME și prima încercare de la Sun de a oferi un mediu Java pentru dispozitiv cu resurse embedded mult mai limitate. În 1999, recunoscând că "o mărime nu se potrivește tuturor", Sun a anunțat divizare de Java în trei versiuni:

- Java 2 Enterprise Edition (J2EE)
- Java 2 Standard Edition (J2SE)
- Java 2 Micro Edition (J2ME).

Deși MIDP 1.0 generat un entuziasm considerabil în rândul comunitatii Java Wireless, a fost, de asemenea, lumea s-a dat seama că MIDP 1.0 era limitat de capacitățile sale de a accesa funcționalitatea oferite de un smartphone tipic în cadrul unui MIDlet. În consecință, la scurt timp după eliberarea MIDP 1.0, comunitatea Java Wireless a început să lucreze la consolidarea capacităților de MIDP. Acest lucru s-a manifestat în MIDP 2.0 (JSR 118), lansat în forma sa finală în noiembrie 2002, și o gamă de extensie API JSRs, toate făcând parte din Java Community Process. Aceste evoluții oferă o creștere substanțială a funcționalității disponibile pentru MIDlet-uri. În consecință, cea mai recentă versiune de Symbian OS (versiunea 7.0s) și UIQ 2.1 au trecut la un singur concept tehnologic Java bazat pe J2ME CLDC și MIDP 2.0 (API-uri suplimentare, plus J2ME opțional). J2ME MIDP este stabilit, în prezent ca platforma Java omniprezenta în aria de telefonie mobilă și, ca atare, Symbian va continua să evolueze și spori posibilitățile de CLDC/MIDP pe care le ofera. [11]

Să ne uităm la beneficiile principale ale limbajului Java pentru dezvoltarea serviciilor și modul în care acestea corespund nevoilor pieței:

- Securitate: servicii și aplicații ce nu pot fi distruse
- Standardizare: mai mulți dezvoltatori și instrumente înseamnă mai multe ca pot fi dezvoltate servicii
- Robustete: mai puține defecte
- Dezvoltarea rapidă: timp de lansare pe piață mai scurt
- Portabilitate: furnizorii de servicii pot lansa oricât de multe telefoane mobile

Symbian folosește Java pentru a expune avantajele și punctele forte ale Sistem de operare Symbian prin intermediul API-urilor, care, în general, sunt standard. Multe dintre noile API-uri Java Wireless, cum ar fi multimedia, Bluetooth și PIM, sunt interfețe pe funcționalitate nativă, ceea ce înseamnă că o implementare Java este bună doar ca platformă de bază. Mai degrabă decât o implementare vanilla MIDP, Symbian ofera un mediu Java, care va permite operatorilor și furnizorilor de servicii să creeze venituri prin servicii interesante, bogate.

3.1: Java Micro Edition

Java Platform, Micro Edition, sau Java ME, este o platforma Java conceputa pentru sistemele embedded (dispozitive mobile sunt un astfel de tip de sisteme). Dispozitive țintă variază de la controlul industrial până la telefoane mobil. Java ME a fost cunoscut anterior ca Java 2 Platform, Micro Edition (J2ME).

Java ME a fost proiectat de către Sun Microsystems, acum parte a Oracle Corporation; platforma înlocuiește o tehnologie similară, Personal Java. Inițial elaborat în cadrul Comunitatii Java ca

JSR 68, diferite nuanțe de Java ME au evoluat în JSR-uri (Java Specification Request) separate. Sun oferă o implementare de referință a caietului de sarcini, dar a avut tendința de a nu furniza implementări libere binare pentru Java Runtime Environment ME pentru dispozitive mobile, mai degrabă bazându-se pe părțile terțe să furnizeze propriile lor medii.

Proiectat pentru telefoane mobile, Mobile Device Profile Information include un GUI (Graphic User Interface), și un depozit de date API (Application Programming Interface), și MIDP 2.0 include un API de bază pentru jocuri 2D. Aplicațiile scrise pentru acest profil sunt numite MIDlet-uri. Aproape toate telefoanele celulare noi vin cu o punere în aplicare MIDP, iar acum este standardul de facto pentru jocuri descarcabile pentru telefonul mobil. Cu toate acestea, multe telefoane mobile pot executa numai acele MIDlet-uri care au fost aprobate de către carrier, în special în America de Nord.

JSR 271: Mobile Device Profile Information 3 (eliberare definitivă pe 09 decembrie 2009) a specificat treia generație Mobile Information Device Profile (MIDP3), extinzându-și funcționalitatea în toate domeniile, precum și îmbunătățirea interoperabilității între dispozitive. Un obiectiv cheie al designului de MIDP3 este compatibilitatea cu MIDP.

Sun a creat Java ME cu scopul de a putea rula pe echipamente cu memorie redusă, grafică limitată, care să poată fi compatibil cu mai multe sisteme de operare. Ei au creat mașina virtuală Kauai (KVM), ce necesită 10% din resursele cerute de mașina virtuală Java (JVM).

Rezultatul final a constat în construirea Java ME din trei componente de nivel înalt, ce satisfac o gamă largă de soluții:

- 1. Configurații**
- 2. Profiluri**
- 3. Pachete optionale**

Dintre facilitățile Java SE (Java Standard Edition) la care s-a renunțat în Java ME sunt:

- Reflectia
- Grupuri de thread-uri și controlul avansat al thread-urilor
- Loader-e pentru clase definite de utilizatori
- Interfața nativă Java
- Finalizarea automată a obiectelor

3.1.1. Configurații

Configurațiile definesc sunt cea mai de bază asemănare între categoriile de dispozitive care în ceea ce privește memorie alocată, puterea procesorului și conectivitatea în rețea. O configurație este caietul de sarcini al unei JVM și un set de bază de biblioteci de clasă necesare care sunt în mare parte versiuni simplificate ale echivalentelor desktop. Există în prezent, doar două configurații:

- The Connected Limited Device Configuration (CLDC): a avut ca scop în mod specific telefoane mobile, pagerele cu două sensuri și PDA-urile low-level.

- Connected Device Configuration (CDC): a avut drept scop dispozitive cu mai multa memorie, conectivitatea la rețea mai buna și procesoare mai rapide.

Majoritatea telefoanelor nu folosesc configuratia CDC, desi exista unele telefoane cu sistem Symbian ce au o platforma Java bazata pe CDC. Cele mai multe telefoane folosesc aplicatii MIDP/CLDC.

Până în prezent au existat două versiuni ale CLDC, cea mai recentă fiind CLDC 1.1. Majoritatea telefoanelor mobile livrate începând cu anul 2003 conțin versiunea 1.1 de CLDC. CLDC 1.1 include, printre alte actualizări, suport pentru operații în virgula mobilă prin introducerea de clase float și double, numiri de thread-uri și întreruperi. Cerința minimă de memorie a fost, de asemenea, ridicată de la 160 KB la 192 KB.

3.1.2. Profiluri

Profilurile sunt nivelul superior unei configurații și se lasă să fie adaptate și specializate pentru o anumită clasă de dispozitive într-o piață în continuă creștere, cum ar fi telefoanele mobile. Un profil definește în mod eficient un contract între o aplicație și un set de dispozitive de același tip, de obicei, prin includerea bibliotecilor de clase care sunt mult mai specifice domeniului decât cele disponibile în orice configurare specială.

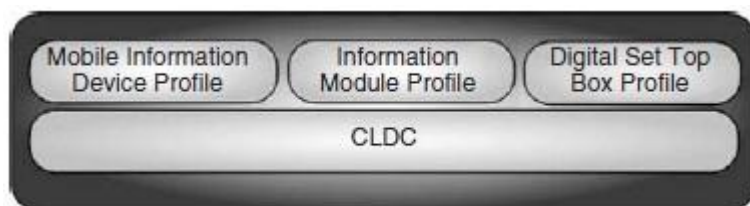


Figura 2. Profilurile CLDC

Figura 2 prezintă cele trei profiluri care se extind în prezent CLDC. Mobile Information Device Profile (MIDP) este cel mai răspândit. Information Module Profile (IMP) are ca țintă dispozitive mici care poate nu au nici o interfață, cum ar fi aparatele de parcare, cabinele telefonice, și automatele de mâncare. Digital Set-Top Box Profile are ca țintă piața televiziunii prin cablu.

Aplicațiile care utilizează bibliotecile MIDP sunt numite MIDlet-uri. MIDP definește funcționalitatea de bază pe care aplicațiile mobile o pot stăpâni (de exemplu, interfața cu utilizatorul (UI), stocarea locală și conectivitatea rețelei) și precizează, de asemenea, managementul ciclului de viață al aplicației. Versiunea originală MIDP a fost începută în noiembrie 1999. Chiar dacă nu beneficia de biblioteci pentru jocuri, asta nu s-a oprit sute de titluri fiind scrise și vândute la nivel mondial. De fapt, atât de mult succes a avut MIDP 1.0, încât în anul 2001 s-a început lucrul la a doua versiune - MIDP 2.0.

Bibliotecile MIDP 2.0 include un API UI de nivel înalt cu suport pentru widget-uri standard (celule text, checkbox-uri, butoane, etc), precum și un API UI de nivel scăzut ce suporta contexte grafice destinate, în principal, dezvoltării de jocuri. API de nivel scăzut include rutine standard

pentru crearea de primitive grafice (linii, poligoane, arcuri, etc), precum și clipping, pânze full-screen, regiuni, manipulare de imagini, fonturi, pensule, culori, si buffer-e off-screen. Suport pentru persistența locală asupra MIDP este furnizat prin sistemul de management al înregistrărilor (Record Management System-RMS). Framework-ul generic CLDC fost extins cu conexiuni suplimentare, cum ar fi HTTP, TCP și UDP.

3.1.3. Pachete optionale

API-urile optionale de uz general oferă un set de funcționalitate, care este nu este specific pentru orice tip de dispozitive și au permis platformei Java ME să evolueze și să adopte tehnologii emergente. Ele adaugă funcționalități suplimentare la un profil și, uneori, sunt încorporate în profil, pe măsura ce tehnologia se dezvoltă. Grafica 3D a aparatelor mobile pentru J2ME API (JSR-184) și Location API (JSR-179) sunt exemple bune de pachete opționale bune. Este comun de a utiliza pur și simplu termenul de "JSR" (Java Specification Requests) atunci când ne referim la un set de pachete pe care un dispozitiv le poate suporta, cu toate că configurațiile și profilurile sunt produse ale JSR-uri particulare.

Java ME a avut un start cu hopuri - realitățile și presiunile concurenței, în mare parte datorate creșterii fenomenale a industriei de jocuri mobile, a condus la o mare disparitate în suportul de JSR opționale în întregul set de dispozitive disponibile pe piață. Nu a fost nici un set standard de JSR-uri disponibile, unele implementări nu au respectat cu cerințele JSR existente și unele au oferit conformitate parțială. A fost extrem de greu chiar și pentru a afla care JSR-uri erau suportate de orice receptor. Uneori, nereguli au avut loc în cadrul aceleiași familii de telefoane mobile de la același producător, cu API-uri care lipsesc pentru niciun motiv evident. Producătorii ar adăuga, de asemenea, propriile API-uri pentru a permite funcționalitate non-standard sau funcționalități suplimentare care nu a fost acoperite de JSR-urile standard. În timp ce aceasta abordare viza necesitatea pe termen scurt, bibliotecile au trebuit să rămână aceleași din motive de compatibilitate pentru mult timp după ce au fost înlocuite.

Rezultatul net a fost o serie de soluții capabile să faciliteze programarea în aplicațiilor Java ME, în special a jocurilor, care aveau ca țintă sute de dispozitive deodată. Case de dezvoltare jocuri au scris biblioteci întregi încercând să lucreze în aceste medii eterogene și apoi au trebuit să lucreze pentru a recupera investițiilor făcute. Acest fenomen nefericit este cunoscut sub numele "fragmentare" și sintagma "scrii o dată, execuți oriunde" a devenit "scrii o dată, debug-uești peste tot". Acest lucru a avut un efect extrem de negativ asupra timpului până la aducerea pe piață a aplicațiilor Java ME și a jocurilor.

3.2 Java ME vs. Symbian C++

De ce nu folosim doar Symbian C ++? Pentru a răspunde la aceasta e nevoie să ne uităm în primul rând la ceea ce fac oamenii cu smartphone-urile lor și apoi cum se realizează aceste cerințe din punct de vedere tehnic.

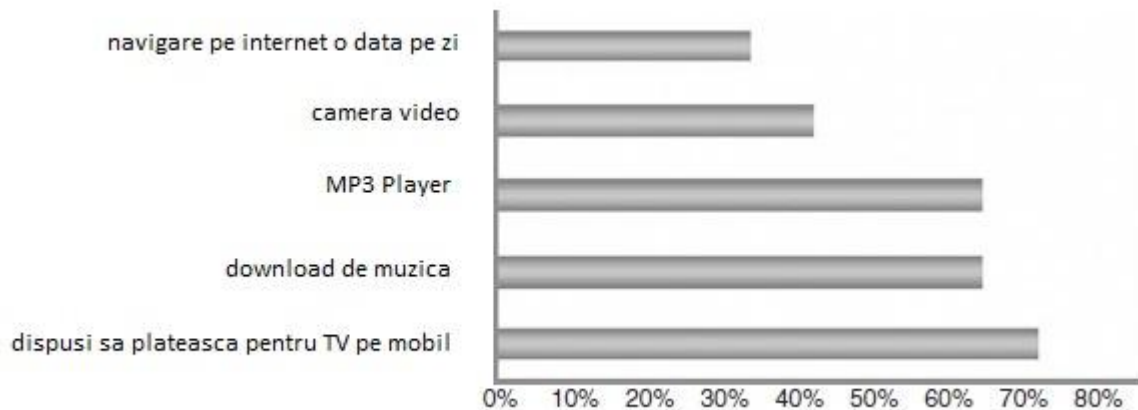


Figura 3. Utilizarea smartphone-urilor

Figura 3 arată rezultatele unui sondaj în ceea ce privește modul în care oamenii sunt dispuși să utilizeze smartphone-urile lor. În plus, dacă ar fi întrebăm un eșantion aleatoriu de oameni pe stradă ceea ce face în mod regulat, ne așteptăm de asemenea ca activitățile lor să includă jocuri, trimiterea de mesaje text, trimitere de imagini, rețele sociale, verificarea scorurilor jocurilor, navigare, vizionarea clipurilor video, verificarea prognozei meteo sau a traficului.

Este clar că există multe utilizări pentru smartphone-uri, dintre care majoritatea implică descărcarea de conținut pe mobil, cum ar fi aplicații de instalat și jocuri, achiziționare de multimedia, cum ar fi muzica, video, TV pe mobil, sau pur și simplu descărcarea de pagini web în timp ce navigați pe Internet. "Conținutul" este cuvântul cheie. O platformă mobilă high-end nu poate exista fără conținut. În absența acestuia, nu este nimic care să convingă utilizatorul să cumpere un dispozitiv high-end. Cine ar cumpăra un smartphone complet echipat pentru a face apelurile de voce și trimite numai mesaje scrise?

În teorie, este posibil să se limiteze tot conținutul instalabil doar la cel creat utilizând Symbian C++, în scopul de a controla sursa acestuia. Dar acest lucru ar rezulta în ingradirea posibilităților, ceea ce este contrar principiului Symbian OS ca platformă deschisă, deoarece milioane de ingineri talentați care sunt nu este bine instruiți în Symbian C++ nu pot contribui la platforma, precum și orice conținutul existent ce nu este scris în C++ Symbian nu poate fi reutilizat.

Atunci când dezvolti programe folosind Java ME și Symbian OS, combini liderul mondial în sistem de operare smartphone cu cea mai de succes platformă de dezvoltare pentru telefoane mobile până în prezent. Symbian OS aduce coerența, performanță, fiabilitate și varietate pentru platforma Java ME, iar pe de altă parte, îi lipsesc cele mai comune limitări ale altor sisteme de operare.

Exista un numar enorm de aplicatii Java ME pe piata, depasind cu mult cele scrise in Symbian C++. Numarul de programatori Java ME pentru mediu mobil este de asemenea in crestere si, fara indoiala, mai mare decat programatorii de Symbian C++.

Sunt unele cazuri in care este nevoie de a folosi viteza maxima, performante in timp real, control fin sau acces direct la serviciile native, facilitati ce nu pot fi atinse in Java. Cand se doreste putere de nivel redus, integrare stransa cu sistemul de operare sau feature-uri avansate ce nu sunt acoperite de Java, cea mai buna varianta este Symbian C++. De exemplu, serviciile nativee Symbian OS iti permit sa dezvolti "carlige" in stiva TCP/IP si sa adaugi extensii componente de mesagerie Symbian OS. Java ME nu este designat pentru asemenea cerinte. [3]

Capitolul 4. Comparatie cu alte sisteme de operare mobile

Android este câștigătoare piața sistemelor de operare, prin performanțele sale și performatele foarte bune din sfera tehnologica. Se preconizeaza ca va intrece iOS 4 și Windows 7 de la Apple și Microsoft, respectiv. Aceasta permite utilizatorului să aiba o privire prin lucruri împreună cu un sprijin hardware. Sistemul de operare Android preia spațiul mobil. Intrarea discreta pe a Android pe piata de telefonie i-au castigat faima imediata si reusit vanzari mai multe decât veniturile aduse de iPhone.

iOS, sistemul de operare iPhone de la Apple a asistat de performanța îmbunătățită a produselor Apple, printre care iPad, Apple TV si iPod touch, împreună cu adăugarea unei colecții uriașe de aplicatii în sistemul de operare. Punctul forte sistemului de operare Android, comparativ cu iOS este caracteristica sa deschisa, care o deosebeste de iOS. Această caracteristică permite Android cel mai bun acces la internet. iOS este un rival al Android, care si-a castigat suficienta reputata pe piața tehnologica.

Diferenta dintre Symbian si Windows Mobile este foarte vizibila. Se pare că ambele pot îndeplini sarcinile rudimentare esențiale legate de multimedia, dar diferenta consta in modul lor de lucru. Ambele au unele elemente pozitive și negative pe care le separa. Symbian oferă standard deschis care atrage utilizarea diverselor aplicații, pe de altă parte, Microsoft oferă un mediu protejat pentru utilizatorii săi în ceea ce privește software-ul. Diferența de configurare este una majora la amandoua. Microsoft a crescut propria platforma de dezvoltare, sub forma de .NET și Symbian pe de altă parte, are de asistență instrumentelor Java și alte metode, dar acestea nu prea sunt bine stabilite ca Java încă.

Se poate concluziona despre Symbian și Windows Mobile că amandoua tipologiile ofera topografii la comun, iar amandoua au interfete multimedia similare, deși provocarea propusă de Microsoft în ceea ce privește smartphone sale nu poate fi compensată, deoarece crede în competențe sale.

Toate sistemele de operare au generat concurență acerbă pentru fiecare dintre ele. Toate acestea concureaza și au succes în termeni de funcții, performanță, caracteristici și versatilitate de

aplicații. Android și iOS sunt amandoua bine configurate și spontane. Competiția dintre sistemele de operare smartphone devine agresiv pe zi ce trece, iar câștigătorul este cel care anticipează tendințele ascunse ale tehnologiei și aduce caracteristici noi și excepționale în ofertele lor. [13]

3.1 Symbian vs. Android

Când vine vorba de sistemele de operare mobile, există două opțiuni majore de luat în considerare: Symbian și Android. Amandoua au specificitățile și defectele lor. Symbian OS a lansat prima variantă în 2005 (Symbian v9). Aceasta a avut numeroase update-uri, printre care Symbian Belle este ultima ediție. Android, pe altă parte, a fost lansat cu câteva luni înainte de Symbian. A fost cumpărat de Google în 2005 și a lansat diferite serii începând cu 2008.

Cele două sisteme de operare, împreună cu creatorii acestora (Nokia și Google) au încercat să se învingă unul pe celălalt în ceea ce privește performanța și serviciile. Poate doar pasionații de jocuri au abilitatea de a determina un câștigător, din moment ce amandoua tind să aducă o experiență superbă jocurilor.

Sistem de operare deschis/închis

Android este un sistem de operare de tip open source, dar cu o platformă de dezvoltare închisă, în timp ce Symbian este open source de tip out-n-out.

Feature-uri și instrumente

Symbian suportă programare C++. Are totuși și suport Java de nivel înalt, astfel oferind o experiență pentru jocuri foarte bună. El afișează jocuri cu ușurință, fără niciun sistem de buffering, ca în majoritatea jocurilor on-line. Din păcate, acest sistem de operare nu funcționează la fel de bine când vine vorba de descărcat jocuri.

Android are o gamă largă de feature-uri, incluzând framework de aplicații, mașina virtuală Dalvik, grafică optimizată, accelerometru, busolă, GPS și diverse aplicații precum un program SMS, hărți, calendar, client de e-mail, contacte și programare Java. De asemenea suportă videoclipuri, imagini, muzică, și alte formate mai complicate. Cu toate acestea, Android are un suport foarte slab pentru jocuri Web, făcându-l nefezabil pentru jocuri on-line.

Utilizabilitate

Symbian si Android sunt amandoua usor de folosit si dispun de o interfata simpla. Cele doua sisteme de operare vin cu o experienta in ceea ce priveste jocurile foarte buna, dar Symbian este de preferat pentru jocurile on-line.

Rețele

Telefoanele pe baza de Android au suport pentru Internet wireless, adica utilizatorii pot intra pe Web de pe telefon, fara sa aiba conectie la Internet la telefoane. Symbian, pe de alta parte, nu are un accelerator de retea pentru o video-telefonie de calitate. Dar utilizatorii pot downloada un astfel de accelerator pentru a spori conexiunea la Internet.

Diferente hardware

Smartphone-urile pe baza de Symbian, ca Nokia N8 au anumite erori cand incearca sa deschida fisiere web de dimensiuni mari. Interfata hardware a Android-ului dispune de camera cu autofocus, callback pe eroarea de camera, o interfata ce ofera o imagine de la poza captata, campul geomagnetic si senzor.

Internet

Browser-ul Symbian poate donwloada cate un fisier odata, dar suporta navigare pe mai multe ferestre. Android 2.0, pe de alta parte, vine cu browser-ul Firefox Mobile, care ofera utilizatorilor viteza mai ridicata si navigare fara buffer.

Multimedia

Atat telefoanele pe baza de Symbian, cat si cele pe baza de Android ofera facilitati multimedia bune. Totusi, imaginile si videoclipurile par sa fie mai de calitate pe telefoanele cu ecrane tactile pe baza de Android. [14]

3.2 Symbian vs. Windows Mobile

Caracteristici generale

Amandoua sistemele de operare au facilitati ca: profiluri configurabile, editoare text, suport avansat pentru navigare web (taste de navigare). Pe langa tehnologii specifice Nokia, Symbian mai beneficiaza de suport pentru mesagerie Fring, aplicatii VoIP. Windows Mobile are totusi o mai buna sincronizare cu PC-ul prin Active Sync, cititor incorporat Braille, acces imbunatatit pentru Windows Media Player.

Aplicatii incorporate

Ambele dispun de cititoare de audiobook-uri si de dictionare. Exista programe care le diferentiaza. Symbian are identificatorul de caractere KNFB Reader si Fring (pentru mesagerie si VoIP). Windows Mobile are Microsoft Office Mobile (Word, Excel, PowerPoint, OneNote), Skype, etc.

Tipuri de telefoane ce suporta Symbian si Windows Mobile

Amandoua sistemele de operare sunt suportate pe telefoane Samsung. Symbian mai poate fi purtat si de telefoane Nokia. Windows Mobile poate fi instalat pe mai multe tipuri de telefoane, dintre care: Asus, Fujitsu Siemens, HP, HTC, Motorola, O2, Palm, Qtek.

Sistem de operare inchis/deschis

Asa cum am spus mai sus, Symbian este un sistem de operare deschis si gratuit. Windows Mobile este un sistem closed source, cu toate aceste interfetele sale sunt open source si bazate pe Microsoft Win32 API (application programming interface). [15]

Capitolul 5. Dezvoltarea unei aplicatii pentru Symbian

In urma cu cativa ania, producatorii de telefoane mobile foloseau propriile lor sisteme de operare. Telefoanele mobile erau destul de simple si nu necesitau sisteme de operare complexe si platforme de dezvoltare complexe.

In figura 4 avem diagrama unei arhitecturi software a unui dispozitiv mobil. Dupa cum se poate vedea, Java si aplicatiile native difera prin faptul ca aplicatiile native au acces direct la sistemul de operare al telefonului si la software-ul sistemului, in timp cee aplicatiile Java au acces limitat prin mai multe API-uri Java.

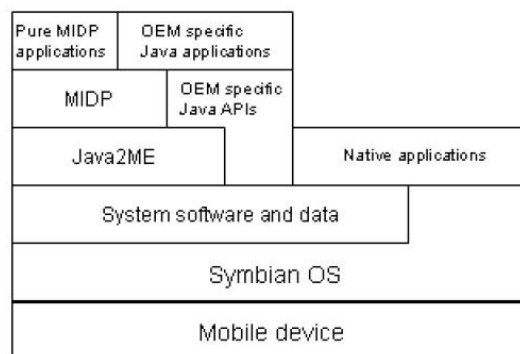


Figura 4. Arhitectura tipică a software-ului unui dispozitiv mobil

5.1 Echipamente de dezvoltare

Producatorii pot oferi instrumente de dezvoltare SDK-uri cu emulatoare, API-ur, librarii si documentatii pentru propriile lor produse. Aceste SDK-uri pot rula pe IDE-uri comune (de exemplu, Carbide.C++). Uneori, interfetele grafice trebuiesc codate manual, facand dezvoltarea de programe mult mai anevoioasa.

In Figura 5 se poate observa arhitectura mediului de dezvoltare si legatura dintre SDK-uri, IDE-uri si API-uri.

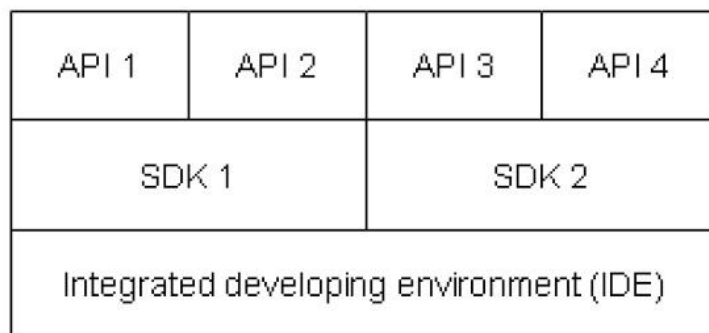


Figura 5. Arhitectura unui mediu de dezvoltare

SDK-urile sunt diferite pentru fiecare tip de telefon. Dintre IDE-uri, se pot mentiona:

- Microsoft Visual Studio IDE
- Borland C++ Mobile Edition IDE
- Metrowerks CodeWarrior Wireless Development Kit for Symbian OS IDE

Cel mai raspandit API este Symbian OS API. API-urile pot fi folosite pe orice dispozitiv cu o versiune corespunzatoare de Symbian OS. Sunt diferite versiuni de API-uri Symbian, ele variind destul de putin.

5.2. Aspecte de luat in seama in dezvoltarea unei aplicatii

Procesorul

Procesoarele RISC nu sunt prevazute cu operatii in virgula mobila. Acest lucru face ca operatiile matematice sa fie lente si se prefera lucrul cu intregi si numere in virgula fixa.

Daca la inceputul sistemului de operare Symbian, dispozitivele pe care acesta rula aveau viteze de maxim 100-200 MHz, astazi se ajunge la viteze de 1GHz.

Memoria

Limitările de memorie produc cele mai mari restricții pentru telefoanele mobile. Există trei tipuri de memorie în telefoanele mobile:

- Read-only: pentru sistemul de operare și pentru sistemul software
- Memorii lente read-write: pentru aplicații și date
- Memorii rapide read-write: pentru aplicațiile în funcțiune

Memoriile de stocare pot fi extinse prin carduri de memorie.

Sunt de preferat formatele de fișiere ce suportă compresie. Dezvoltatorul trebuie să aibă grijă folosește alocarea dinamică a memoriei pentru a preveni „scurgerile” de memorie, care pot duce la epuizarea resurselor. Aceste fenomene pot avea urmări grave, ca: blocarea sistemului și pierderea de date.

Lungimea de bandă pentru comunicații

Există anumite restricții în rețelele GSM și GPRS ce trebuie luate în considerare. În Africa, Europa, Asia și Orientul Îndepărtat, majoritatea furnizorilor GSM de telefonie mobilă folosesc benzi de 900 MHz și 1800 MHz. GPRS permite transferul de date prin servicii ca 2G, 3G și 3.5G cu viteze care au depășit deja 1 Mbps. Rata de transfer este un factor important în aplicațiile ce transferă multe date, cum ar fi cele de tip multimedia. Timpul de răspuns joacă un rol important în aplicații de streaming, chatting sau jocuri multiplayer. Întârzierea este de obicei mai mare (câteva sute de milisecunde) și siguranța conexiei este de asemenea mai slabă în rețelele mobile decât în rețelele fixe tradiționale. Multe telefoane cu Symbian folosesc Bluetooth pentru aplicații wireless de distanță redusă. Bluetooth poate fi folosit pentru comunicații destul de rapide cu calculatorul sau între două telefoane mobile.

Capitolul 6. Bibliografie

1. Harrison, Shackman - "Symbian OS C++ for Mobile Phones Volume 3", 2007
2. Babin – "Developing Software for Symbian OS", 2007
3. Hayum – "Java ME on Symbian OS", 2007
4. Jode – "A developer's guide to MIDP 2.0", 2004
5. Sonera MediaLab - „Symbian Application Development White Paper”, 2003
6. http://en.wikipedia.org/wiki/Symbian_OS
7. http://en.wikipedia.org/wiki/S60_%28software_platform%29
8. http://en.wikipedia.org/wiki/Series_40
9. <http://www.newmobilemedia.com/android-vs-symbian.htm>
10. <http://www.symbian-guru.com/welcome/2010/03/10-things-symbian-does-better-than-android.html>
11. http://www.developer.nokia.com/Community/Wiki/Naming_and_versioning_of_Java_Runtime_for_Symbian
12. http://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition
13. <http://www.tech2date.com/android-vs-ios-vs-symbian-vs-windows-mobile.html>
14. <http://www.gadg.com/2011/09/20/symbian-vs-android-a-brief-comparison/>
15. <http://www.codefactory.es/en/products.asp?id=333>