

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Sumaniųjų telefonų operacinių sistemų analizė

Comprehensive Analysis of SmartPhone OS

Bakalauro darbas

Atliko: Kęstutis Matuliaskas (parašas)

Darbo vadovas: lekt. Vytautas Valaitis (parašas)

Darbo recenzentas: doc. dr. Sigitas Dapkūnas (parašas)

Versija 3.21

Vilnius – 2012

TURINYS

ĮVADAS	5
1. TELEFONAI IR MOBILIOSIOS OS	7
1.1. Mobiliųjų telefonų klasifikacija	7
1.2. Funkcinis telefonas su JAVA aplikacijų paleisties aplinka	7
1.3. Moderniausia funkcinio telefono operacinė sistema.....	10
1.4. Išmanusis telefonas su operacine sistema	10
1.5. Operacinės sistemos išmaniesiems telefonams	11
1.6. Išmaniųjų mobiliųjų telefonų ir jų OS populiarumas ir rinka	12
1.7. Aplikacijų kiekių jų internetinėse parduotuvėse palyginimas	13
1.8. Aplikacijų kūrimo terpės – karkasų aplinkos, jų skirtumai	14
2. MOBILIŲJŲ TELEFONŲ OS HIERARCHIJA IR APŽVALGA.....	17
2.1. Symbian, Maemo (MeeGo).....	17
2.2. Google Android, Samsung Bada.....	18
3. IŠSAMI ESAMŲ IŠMANIŲJŲ TELEFONŲ MODERNIŲ OPERACINIŲ SISTEMŲ ANALIZĖ	21
3.1. Nokia Belle (S60 5.2, Symbian 10.1) operacinė sistema.....	21
3.2. Windows Phone 7	24
3.3. Google Android 2.3.....	27
3.4. Apple iOS 5.1.....	31
3.5. HP/Palm webOS 2.2	34
3.6. Research in motion(RIM) Blackberry OS 7.1	37
4. IŠMANIŲJŲ TELEFONŲ OS PALYGINIMAS	40
4.1. Išmaniųjų įrenginių aparatinės įrangos palyginimas.....	40
4.2. Išmaniųjų telefonų su moderniomis OS funkcionalumo palyginimas	41
4.3. Mobiliosios OS PĮ kūrimo galimybės įvairiose OS su IDE.....	42
4.4. Bendras išmaniųjų OS įvertinimas (1-10).....	43
5. ATEITIES PERSPEKTYVOS IR TENDENCIJOS	45
5.1. Išmaniųjų OS abstrakcijos ir globalumo tendencijos.....	45
5.2. Debesų kompiuterijos ir interneto aplikacijų įtaka išmaniųjų telefonų OS	46
5.3. Išmaniųjų telefonų OS ateities perspektyvos	46
5.4. Išvados ir teiginiai iš išmaniųjų OS rinkos prognozių	47
5.5. Ko tikėtis Symbian ir Series 40 platformų PĮ kūrėjams?	48
5.6. Ko tikėtis Samsung Bada OS PĮ kūrėjams?	50

5.7. Ko tikėtis Windows Phone, Android ir iOS platformų PĮ kūrėjams?	51
5.8. Ko tikėtis Blackberry OS PĮ kūrėjams?	51
REZULTATAI IR IŠVADOS	52
SĄVOKŲ APIBRĖŽIMAI	54
SANTRUMPOS	55
ŠALTINIAI	56
1 priedas. Vizualizavimo testo aplikacijos kodas PhoneGap karkase	65
2 priedas. GUI programinio kodo pavyzdžiai išmaniųjų OS-ose	67

IVADAS

Šio bakalauro darbo **tikslas** – ištirti ir palyginti įvairias operacines sistemas, naudojamas išmaniųjų telefonų tarpe. Juo siekiama parengti rekomendacijas ir išvadas, kuriomis remiantis, aplikacijų kūrėjai galėtų nuspręsti, kokią platformą pasirinkti ir koku būdu kurti aplikaciją, t.y. kurią iš 3 išmaniųjų telefonų aplikacijų kūrimo aplinkų naudoti: (1) SDK, pateiktą kartu su platforma (native), (2) multi-platforminius web-karkasus, sukurtus naudojant „native“ SDK ar (3) globalią HTML5 kalbą, javascript karkasus ir CSS 3.0 vizualizacijos technologijas [Yas12, 5 p.].

Išmaniųjų telefonų OS rinka šiuo metu kinta žaibiškai, lyderiai praranda aiškias pozicijas, iškyla sunkiai suprantami programiniai produktai. Šiais laikais OS sistema apskritai praranda esmę, nes daugumai ji pradeda tapti tik terpe pasiekti interneto naršyklę, kuri visose OS yra standartizuojama. Šio bakalauroinio darbo metu siekta kritiškai ir išsamiai įvertinti esamas mobiliąsias OS (toliau – MOS), jas palyginti išskiriant trūkumus ir privalumus. Taip pat įvertinti ir palyginti būsimas OS bei padaryti išvadas apie šių OS praktiškumą, panaudojamumą ir tikėtinas perspektyvas.

Darbo rašymo metu buvo nagrinėjami kiti panašių tikslų siekiantys straipsniai ir darbai [Yas12], rasti internete IEEE duomenų bazėje ar e-knygose. Buvo išnagrinėti šių darbų trūkumai ir privalumai, o gautos išvados panaudotos bakalauroiniame darbe. Kituose darbuose dažniausiai pasirenkama tirti konkrečia daugelio OS savybę, pvz. spartą [Yas12], konkretų karkasą [ECM+12], globalius įrankius [OT10] arba palyginti dvi mobiliąsias OS [Che10, SGE09, TSL12]. Šiame darbe stengiasi žvelgti į OS tiek konkrečiai (individualūs SDK), tiek plačiąją prasme (multi-platforminiai karkasai), kaip į abstrakčią platformą programuotojams – jos bendrumus ir specifiskumus (konkrečius trūkumus). Taip pat aptartos globalių standartų galimybės – karkasus uždengiantis specifinis OS funkcionalumas, siūlantis bendrus sprendimus programuotojams, pvz., QML platforma arba iOS grafinių, net trimačių, papildinių kūrimą, naudojant HTML5 standartą, multi-platforminį PhoneGap ir kitus.

Bakalauroinio darbo autoriaus praktikos gretutinės užduotys buvo mobiliosios aplikacijos, pavadinimu „Blogink bet kur“, kompanijų Omnitel ir Nokia organizuotame konkurse „Programėlių dirbtuvės“, kūrimas bei prisidėjimas prie iOS aplikacijos kūrimo „Credit Sesame, Inc.“ įmonei, Silicio slėnyje, Kalifornijos valstijoje, JAV, kurioje autorius atliko privalomąją praktiką. Taip pat šioje įmonėje buvo atlikta galimybių analizė ir paruoštas projektas, kaip būtų galima perkelti aplikaciją iš iOS į kitą platformą – Windows Phone 7 OS, stengiantis pasinaudoti

bakalauriniame darbe aptariamų karkasų (angl. framework) funkcionalumu bei minimizuoti reikiamų pakeitimų skaičių perkeliant aplikacijos kodą iš vienos OS į kitą OS.

Autorius, rašydamas šį darbą, ne kartą buvo pakviestas ir dalyvavo, tarptautinių kompanijų Nokia (C), Digia (C), Microsoft (C) ir Omnitel (C) surengtose stažuotėse („Nokia QT Training“), mokymuose, konferencijose („MoMo Latvia discussion & conference“, „MoMo Estonia discussion & conference“, „MoMo Lithuania discussion & conference“, „Windows Azure conference & training“) ir pristatymuose, surengtose Vilniuje, Rygoje, Taline, Mountain View (JAV įsikūrusi viena iš Microsoft būstinių). Apie šias kompanijas plačiau išdėstyta žemiau (1 lentelėje).

1 lentelė. Kompanijos

Kompanija	Nokia ^[Mac12a]	Digia	Omnitel ^[Omn12]	Microsoft ^[Mac12b]
Įkurta	1865 m.	2005 m.	1991 m.	1975 m.
Būstinė	Espoo (Suomija)	Helsinkis (Suomija)	Vilnius (Lietuva)	Redmond, WA (JAV)
Pajamos (2011 pab.)	Apie 38 mlrd. litų	Apie 400 mln. litų	Apie 484 mln. Litų	Apie 201 mlrd. litų
Darbuotojų (2012vid.)	Apie 122 tūkst.	1200	Apie 650	Apie 90 tūkst.
Apie	Nokia mob. prietaisų gamintojas	Turi visas teises į QT programinę įrangą	Didžiausia mobiliojo ryšio bendrovė Baltijos šalyse	„Windows“ šeimos OS kūrėja

Pirmoji darbo kulminacija – Lietuvoje surengtos jungtinės kompanijų Omnitel ir jos partnerės Nokia bei programinės įrangos kūrėjų varžybos, pavadintos „Programėlių dirbtuvės“. Šiame renginyje dalyvavo virš 100 dalyvių, į antrą etapą pateko virš 20 komandų, kur 4 komandos pasidalino 15000 litų prizinį fondą ir mobilųjį telefoną Nokia N8. Šio darbo autorius būtent ir buvo apdovanotas mobiliuoju telefonu Nokia N8. Antroji darbo kulminacija – išleista veikiant „Credit Sesame, Inc.“ įmonės, kurioje autorius atliko praktiką, aplikacija, skirta iOS operacine sistema paremtiems įrenginiams (iPad, iPhone, iTab), bei parengtas projektas kaip perkelti iOS aplikaciją į Windows Phone 7 OS terpę.

Bakalaurinio darbo metu autorius kūrė programinę įrangą mobiliesiems telefonams naudodamas Nokia QT (QT Mobile) (skirtą Symbian Belle, Maemo OS), Google App Inventor (Android SDK) (skirtą Android 3.X OS kūrėjams) ir Microsoft Visual Studio 2010 (WP Developer Tools) (skirtą Windows Phone 7 OS) kūrėjo įrankius.

1. TELEFONAI IR MOBILIOSIOS OS

1.1. Mobiliųjų telefonų klasifikacija

Yra 3 kartų mobilieji telefonai:

1-os kartos – mobilieji telefonai (angl. cell phones) – pirmieji mobilieji telefonai, neturėję galimybės diegti/ paleisti papildomų aplikacijų, telefono pagrindinės funkcijos – skambutis ir SMS žinutės siuntimas. Šių telefonų buvo pagaminta daugiau kaip 3 milijardai. Dauguma jų šiuo metu yra nebenaudojami ar utilizuoti. Šių telefonų jau nebegamina jokia žinoma pasaulinė mobiliųjų telefonų gamintoja.

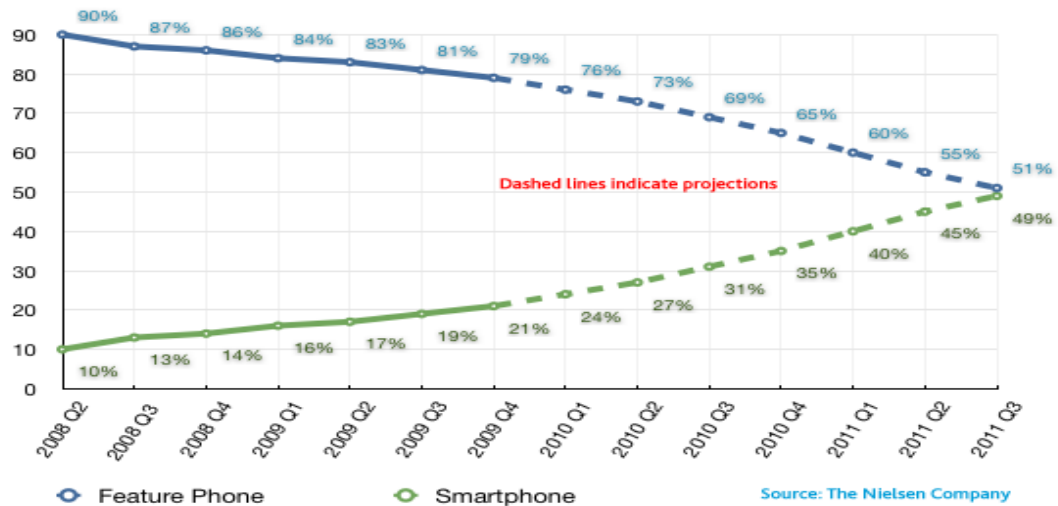
2-os kartos – funkciniai telefonai (angl. feature phones) – tai J2ME ir BREW telefonai. Šiuo metu pasaulyje yra daugiau kaip 2,1 milijardo J2ME palaikančių telefonų, tačiau dėl savo netobulumo ir ribotumo, J2ME pagrindu nėra gaminami jokie šių dienų net vidutinės klasės mobilieji telefonai. Prie J2ME telefonų grupės priskiriama ir Nokia S40 vartotojo sąsaja (UI), kuriai telefonai šių metu dar yra kuriami bei Samsung Bada 1.X, kuri irgi yra laikoma tik funkcinį telefonų OS. BREW retai naudojama ir nepopuliari, nors dar vystoma platforma. Bene žinomiausi BREW platforma paremti gaminiai – Skype telefonai.

3-os kartos – išmanieji telefonai (angl. smart phones) – tai mobilieji telefonai su operacinėmis sistemomis – Nokia S60, Nokia Belle (buvęs Symbian^3), Android, iOS, Blackberry ir kt. Šios kartos telefonų operacinės sistemos skirstomos į dvi grupes – paprastasias (S60, Windows Mobile 6.X) ir sudėtingąsias OS (Nokia Belle, Android 2.X, iOS4, Windows Phone 7). Šiuo metu tai itin sparčiai auganti telefonų rinka ir prognozuojama, jog 2016-ais metais pasaulinėje rinkoje turėtų būti virš 1 milijardo išmaniųjų telefonų [Cam12].

1.2. Funkcinis telefonas su JAVA aplikacijų paleisties aplinka

Šie telefonai yra sumaniųjų telefonų pirmtakai. Kuriant sumaniųjų telefonų platformas – ir mobiliųjų telefonų, ir operacinės programinės įrangos kūrėjai, stengėsi išspręsti visas ar bent jau daugumą problemų, su kuriomis buvo susidurta J2ME funkcinį telefonų platformoje – pvz. aplikacijų sklaidos pasauliniu mastu galimybės ir kt.. Šios kartos mobilieji telefonai pasižymi galimybe paleisti JAVA pobūdžio aplikaciją be instaliacijos į telefono sisteminį branduolį, sukurtą J2ME SDK kūrėjo įrankiais. Šių telefonų rinka pasaulyje šiuo metu yra pati didžiausia, bet dauguma šių naudojamų telefonų yra 3 pasaulio šalyse (Afrikos žemyne, Kinijoje, Indijoje ir pan.). Ten ji sudaro apie 90 proc. rinkos dalies. Europoje, Japonijoje, JAV (žr. 1 pav.) išmaniųjų ir funkcinį telefonų turimos rinkos dalys jau šiomet turėtų susilyginti.

U.S. Smartphone Penetration & Projections



1 pav. JAV išmaniųjų telefonų ir funkcinių telefonų rinkos pasidalijimas,
(C) The Nielsen Company, 2009-11, [Cam12]

Nors šių telefonų rinka ir yra didelė, sukurti vieną aplikaciją šiems telefonams yra labai sudėtinga arba neįmanoma. Šią problemą itin išsamiai aptarė tarptautinės „Nutiteq“ kompanijos atstovas Jaak Laineste, „MoMo Estonia“ seminario, pavadinimu „Mobiliosios PĮ kūrimas – po 3 metų (angl. „Mobile Software Development - 3 years later“) Taline metu, 2011 m. balandžio mėnesį. Savo pranešime, pavadinimu „Žvelgiant į kūrimą J2ME ir BlackBerry (angl. „Insight into J2ME and BlackBerry Development“), jis aptarė problemas, su kuriomis susiduria įmonės kurdamos J2ME aplikacijas. Remiantis ta informacija, bei įvairiais interneto šaltiniais, autorius pateikia šias priežastis:

- Dėl itin didelio nesuderinamumo (angl. incompatibility) tarp skirtingų mobiliųjų telefonų naudojamų JRE („Java Runtime Environment“) versijų.
- Dėl skirtingų, vienu mokamų, kitų nemokamų, vartotojo sąsajų, tarp skirtingų kompanijų.
- Dėl itin skirtingo tų pačių telefonų funkcionalumo realizacijų ir programuotojo prieigos prie tų funkcionalumų.
- Dėl skirtingos turimos telefonų operatyvios atminties kiekio (nuo 100kiB iki 5-6 MiB) ir skirtingų skaičiavimo procesoriaus parametrų, taip pat skirtingų grafinių bibliotekų palaikymo.
- Dėl skirtingų šių telefonų ekranų dydžių ir jų proporcijų bei krypties ir abstraktaus modulio, apimančio visa, nebuvimo.
- Dėl skirtingų įvesties klavišų kiekio ir paskirties, bei abstrakcijos modulio nebuvimo.
- Dėl itin komplikauto arba neįmanomo šių aplikacijų atnaujinimo

- Dėl nesančios galimybės įtraukti reklamas ir kitas daugialypės terpės (angl. multimedia) funkcijas (RSS naujienų srautus ir pan.).
- Dėl globalios (pasaulinės) aplikacijų atsissiuntimo/pirkimo sistemos nebuvimo (tokių kaip „Nokia Store“, „App store“, „Android Market“ ir kt.) – visi atsisiuntimai turėjo būti daromi pačių vartotojų surastuose WAP tinklapiuose (o pati WAP sistema taip pat prarado populiarumą) ir tokių WAP tinklapių mažo kiekio bei ribotumo.

Dėl visų šių priežasčių sukurti aplikaciją, kuri tinkamai veiktų bent jau daugumoje J2ME pagrindu sukurtų telefonų, yra praktiškai neįmanoma. Nėgana to, taip pat fiziškai nėra jokių būdų pateikti savo sukurtą aplikaciją pasaulinei J2ME funkcinėms telefonų rinkai dėl globalaus aplikacijų pirkimo ir parsisiuntimo tinklo, tinkamo šiems telefonams, nebuvimo. Todėl iki 2008 metų aplikacijos kūrimas J2ME platformos telefonams buvo:

- labai brangus – reikėjo parengti kartais iki 20-30 skirtingų programos versijų, kad programa galėtų naudotis bent daugiau nei pusė J2ME platformos naudotojų,
- reikalaujantis didelių finansinių ir proto išteklių – programuotojai turėjo nusimanyti apie daugybės telefonų specifikacijas ir funkcionalumus, o tokių programuotojų įdarbinimas yra labai brangus, be to net pačių elementariausių aplikacijų kūrimo procesas dėl suderinamumo problemų buvo itin lėtas,
- reikalaujantis daugybės žmogiškųjų resursų šias aplikacijas platinant pasauliniame tinkle.

Ir tik 2008-2009 metais kompanija „Sun Microsystems“ (kuri šiuo metu yra kompanijos „Oracle“ dalis ir kuriai priklauso visos teisės į „Java ME“ platformą), padarė keletą svarbių, bet esminių pakeitimų: visų J2ME platformos telefonų JRE („Java Runtime Environment“) versija buvo griežtai apribota iki 1.3 versijos. Tai nebeleido programuotojams rašyti programinio kodo su funkcijomis iš JRE 1.4, 1.5, 1.6 ir dabartinės naujausios JRE 1.7 versijos. Taip pat buvo sukurta „J2ME JSR Mobile 3D Graphics 1.0“, o šiuo metu baigiama kurti 2.0 API kūrėjo sąsaja, kuri žymiai supaprastino žaidimų ir trimačių vaizdų programavimą J2ME platformoje.

Nors šie pakeitimai ir neišsprendė daugumos problemų, tačiau bent kelis kartus supaprastino aplikacijų kūrimą šiai platformai, dėl ko sumaniųjų telefonų operacinių sistemų ir taikomųjų aplikacijų kūrėjai ir telefonų gamintojai J2ME rinką praranda kur kas lėčiau nei norėtų. Norint kurti aplikacijas J2ME platformoje, šiuo metu yra rekomenduojami šie programiniai kūrėjų paketai:

- Java ME SDK (3.0),
- NetBeans 7.0 Mobility Pack,
- Eclipse Mobile Tools for Java.

1.3. Moderniausia funkcinio telefono operacinė sistema

Funkcinių telefonų seriją išplėtė kompanijos Nokia sukurta Series 40 (S40) vartotojo sąsaja, kuri labai standartizavo J2ME platformoje buvusių skirtumų. Tai kartu buvo ir pirmas žingsnis, neužilgo privedęs prie išmaniųjų telefonų operacinių sistemų sukūrimo.

Skirtingai nei Series 60 operacinė sistema, S40 neveikia kaip Symbian branduolio karkasas, bet yra uždaro kodo, ir tik Java taikomasias programas leidžianti paleisti operacinė sistema, neturinti realaus trečiųjų šalių PĮ kūrėjo sąsajos palaikymo (angl. no 3-rd party native API). Tai esminis faktas, dėl kurio ši operacinė sistema nėra vadinama tikrąja OS ir šia OS paremti telefonai yra priskiriami prie funkcinių telefonų, o ne prie išmaniųjų telefonų. Nors vizualinis skirtumas tarp S60 ir S40 yra minimalus, tačiau architektūriškai tai visiškai skirtingos OS, pradedant tuo, kad S40 aplikacijos rašomos JAVA aplinkoje, o ne QT C++ ir jos karkasų aplinkose, kas būdinga Symbian platformos OS (S60v3, S60v5 (Symbian^1), Symbian^3).

Taip pat skirtingai nei S60 serijoje, S40 serijos mobilieji telefonai nepalaiko aparatūrinio kelių užduočių vykdymo vienu metu (angl. hardware multi-tasking). Tačiau, nors ir nėra pilnavertė OS, S40 turi išties didelį funkcionalumą iš pačios OS pusės ir naujausioji S40v6 palaiko tokias funkcijas [Mob12a]:

- WebKit naršyklę su HTML 4.01, CSS2, JavaScript 1.5 ir Ajax palaikymu.
- FlashLite
- Interneto telefoniją (VoIP), POP3, IMAP protokolus.

Verta paminėti, kad kompanija Nokia buvo išleidus ir Series 30 vartotojo sąsają, tačiau išleidusi kelis telefonus, šią sąsają nutraukė, ir naujesnius modelius leido jau Series 40 sąsajoje.

Norint kurti aplikacijas Series 40 platformoje, reikia įsidiegti vienai iš dviejų pagrindinių JAVA integruotų kūrimo aplinkų (IDE) – NetBeans arba Eclipse, pritaikytą Series 40 PĮ kūrėjo paketą (SDK):

- Series 40 SDK for NetBeans IDE 7.0,
- Series 40 SDK for Eclipse IDE.

1.4. Išmanusis telefonas su operacine sistema

Išmanusis telefonas (angl. smartphone) – tai mobilusis telefonas su operacine sistema. Pagrindiniai skirtumai nuo įprasto mobiliojo telefono yra susiję su sudėtinga operacine sistema, kuri: leidžia instaliuoti papildomas programas išplečiančias išmaniojo telefono galimybes; duoda

programuotojui lengvai ir paprastai kurti taikomas programas (angl. apps) šiems mobiliams telefonams; suteikia galimybę programuotojui pasiekti sisteminės aparato funkcijas, tokias kaip SMS žinutės, skambučiai, GPS koordinatės, liečiamasis ekranas (magnetiniai reiškiniai, aplinkos šviesumas (apšvieta)) ir kt.

Pirmuoju išmaniuoju telefonu yra laikomas 2000-aisiais metais išleistas Sony Ericsson R380 mobilusis telefonas, turėjęs lietimui jautrų ekraną ir sujungęs mobilaus telefono ir asmeninio skaitmeninio pagalbininko (angl. PDA) funkcijas. Jis buvo paremta UIQ sąsaja Symbian platformoje, kuri vėliau – 2009 m. buvo pervadinta ir prijungta prie bendros S60v3 sąsajos (Symbian^1).

1.5. Operacinės sistemos išmaniesiems telefonams

Išmaniųjų telefonų bumas prasidėjo su kompanijos Nokia finansuotos organizacijos „Symbian Foundation“ (pastarosios veikla buvo nutraukta 2010 metų antroje pusėje) sukurta Series 60 (toliau – S60) operacine sistema, kurios naujausia versija šiuo metu yra S60v5. Taip pat buvo išleistos S80 ir S90 platformos, bet jos arba buvo nutrauktos arba prijungtos prie S60. Google Chrome ir MeeGo (Nokia) operacinės sistemos yra skirtos internetiniams kompiuteriams (angl. Netbooks) ir e-knygoms (angl. TabletPC), todėl šiame darbe nebus nagrinėjamos.

Kadangi S60 ir Windows Mobile platformų operacinės sistemos buvo šių dienų mobiliųjų telefonų sudėtingųjų operacinių sistemų pirmtakės, tai jas išskirsime į atskirą grupę ir vėliau plačiau nebeaprašinsime, nes jos nebeatitinka šių dienų pilnavertės mobiliosios operacinės sistemos funkcionalumo apibrėžimo. Žemiau pateikiamose 2-oje, 3-ioje ir 4-lentelėse pateikta OS, kurios bus išsamiai nagrinėjamos šiame darbe, taip pat tos OS, kurios bus trumpai apžvelgiamos, bet nėra laikomos pilnavertėmis moderniomis išmaniųjų telefonų OS, bei tos OS, kurios bus tik paminėtos darbo metu (nes jos skirtos ne išmaniesiems telefonams).

2 lentelė. Išsamiai nagrinėjamos modernios OS

Savybė	iOS	Android	Windows Phone	BlackBerry OS	Symbian^3 (Belle)	webOS
Vystytojas	Apple	Google	Microsoft	RIM	Nokia	HP/Palm inc.
Naujausia versija	5.1.1	4.0.4	7.10.X	7.1	10.1	2.2
Branduolys	Mac OS X	Linux	Windows CE 7	Mobile OS	Mobile OS	Linux
Palaikomi CPU	ARM	ARM, x86	ARM			ARM, x86
Progr. kalba	Objekt. C	Java	C#, .NET, XNA	Java	C++	C
Licenzija	EULA, uždaro kodo	Atviro kodo, nemokama	Uždaro kodo, mokama		Atviro kodo, nemokama	Atviro kodo, nemokama

3 lentelė. Nenagrinėjamos 1-os kartos/ funkcinės OS

Savybė	Bada	Windows Mobile	Symbian^1	Maemo
Vystytojas	Samsung	Microsoft	Nokia	
Naujausia versija	1.2; (2.0.5 – nėra naud. įr.)	6.5.3	9.4 (S60v5)	5.0
Branduolys	Linux	Windows CE 5.2	Mobile OS	Linux
Palaikomi CPU	ARM	ARM		
Progr. kalba	C++	C++		
Licenzija	Uždaro kodo, mokama	Uždaro kodo, mokama	Atviro kodo, nemokama	

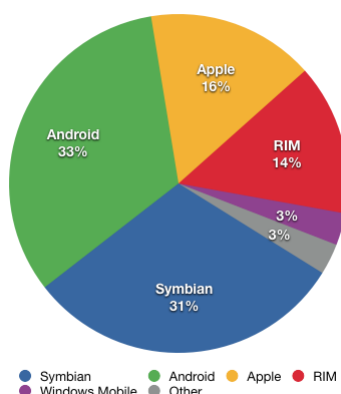
4 lentelė. Netbook OS, TabletPC OS ir TouchPad OS

Savybė	Chrome OS	MeeGo	webOS
Vystytojas	Google	Nokia	HP/Palm inc.
Paskirtis	Netbook	TabletPC	TouchPad
Naujausia versija	0.19	1.2.0	3.0.5
Branduolys	Unix	Linux	Linux
Palaikomi CPU	ARM, x86	ARM, x86	ARM, x86
Progr. kalba	C++	C++	C
Licenzija	Atviro kodo, nemokama	Atviro kodo, nemokama	Atviro kodo, nemokama

1.6. Išmaniųjų mobiliųjų telefonų ir jų OS populiarumas ir rinka

Žemiau pateiktoje 5 lentelėje matome išmaniųjų mobiliųjų telefonų pardavimus 2010-2011 metais su 2012-2015 metų prognozėmis. Virš jos pateiktoje diagramoje (žr. 2 pav.), matome kitos kompanijos – Canals pateiktą rinkos pasiskirstymą 2010 4-ketvirtyje.

2 pav. Išmaniųjų telefonų pardavimai galutiniams klientams (C) Canals, 2010 Q4



5 lentelė. Pasaulinis telefonų su išmaniosiomis OS pardavimas vartotojams (x 1000), (C)

Gartner, 2011-04 / Strategy Analytics, 2012-02 [Mob12b, Spe12], [Spe12]

OS	2010	2011	2012	2015
Symbian (mln.)	111,6	80,1	32,7	0,661
Rinkos dalis (%)	37,6	16,4	5,2	0,1
Android (mln.)	67,2	237,7	310,1	539,3
Rinkos dalis (%)	22,7	48,8	49,2	48,8
Blackberry (mln.)	47,5	51,4	79,3	122,9
Rinkos dalis (%)	16,0	10,5	12,6	11,1
iOS (mln.)	46,6	93,1	118,8	190,0
Rinkos dalis (%)	15,7	19,1	18,9	17,2
Microsoft (mln.)	12,4	26,3	68,2	216,0
Rinkos dalis (%)	4,2	5,6	10,8	19,5
Kitos OS (mln.)	11,4	18,6	21,4	36,1
Rinkos dalis (%)	3,8	3,8	3,4	3,3
Viso: (mln.)	296,6	487,7	630,5	1104,9

1.7. Aplikacijų kiekių jų internetinėse parduotuvėse palyginimas

Aplikacijų parduotuvių rinkoje taip pat didelė konkurencija – tai matome 6-oje lentelėje.

6 lentelė. Aplikacijų parduotuvių populiarumas^{[Wik12], [App11], [Mas11]}

Pavadinimas	App Store	Nokia Store (buvo Ovi)	Android market	Windows marketplace	RIM App World	Samsung Apps
Įkūrėjas	Apple	Nokia	Google	Microsoft	RIM	Samsung
Pradėjo veiklą	2008-07-10	2009-05-26	2008-10-22	2010-10-21	2009-04-01	2010-06-01
Įrenginių (aplikacijoms)	194 mln. ^[Yin11]	825 mln. [Poc10]	100 mln.	3 mln.	46 mln.	5 mln.
Aplikacijų (2011 gegužė)	500 tūkst. ^[Sta12] (2012 birželis)	54 tūkst.	200 tūkst.	100 tūkst. ^[Sta12] (2012 birželis)	30 tūkst.	5,5 tūkst.
Parsisiuntimų	10 mlrd.	0,71 mlrd.	4,5 mlrd.	0,08 mlrd.	0,55 mld.	0,11 mlrd. ^[Big11]
Pelno dalis kūrėjams	70%	70%	70%	70%	70 %	70%

Pirmoji aplikacijų parduotuvė (mokamų ir nemokamų) įkūrė kompanija „Apple“. Ji šiuo metu užima lyderio pozicijas. Tačiau pagal aplikacijų parsisiuntimų skaičių ją po truputi vejasi „Google“ ir jos sukurta „Android Market“ programėlių parduotuvė.

1.7.1. Nemokamos aplikacijų parduotuvių alternatyvos

Autorius turėjo galimybę pabendrauti, su Silicio slėnyje, JAV įsikūrusios, bet iš Lietuvos kilusios, didžiausios pasaulyje nemokamos programėlių mobiliems telefonams talpyklos – „GetJar, Inc.“ darbuotojais – GetJar administratoriumi (angl. System engineer) Arnu Valeika (toliau – A.V.), ir pardavimų operacijų vadybininku (angl. Sales Operations Manager) Romanu Zontovičiumi (toliau – R.Z.). Šiuo metu GetJar sparčiai vejasi kitas, unifikuotas specialiai tik konkrečioms OS skirtas aplikacijų parduotuves. Anksčiau GetJar buvo tik funkcinė telefonų aplikacijų bazė, tačiau šiuo metu tai tapo multi-platformine, visiškai nemokama aplikacijų baze, kuri remiantis tam tikrais algoritmais, kaip teigė A.V., atranka ir iškelia į priekį tas populiarias aplikacijas, kurios įprastose aplikacijų parduotuvėse, nepatenka į pirmus puslapius, taip suteikiant galimybę pasiekti unikalų, kitokį turinį, bei kartu skirtą daugeliui platformų – šiuo metu tai Java ME, BlackBerry, Symbian, Windows Mobile ir Android OS skirtos aplikacijos [Bai10] – o pačia, virš 350 tūkst. aplikacijų turinčia parduotuve [Ind10, Rea09], naudojasi virš 15 mln. Žmonių, kurie aplikacijas parsisiuntė jau beveik 2,5 mlrd. kartų, t.y. apie 3 mln. parsisiuntimų kasdien [59, 61]. Pati GetJar, dėl savo bene 8 metų gyvenimo trukmės (įmonė įkurta 2004 Ilja Laurso, 2012-aisias perleidusio CEO vietą amerikiečiui Chris Dury), žengia antroje vietoje pagal turimą aplikacijų skaičių, iškart po „Apple App Store“.

1.8. Aplikacijų kūrimo terpės – karkasų aplinkos, jų skirtumai

Norint pasirinkti tinkamą kūrimo karkasą, programuotojas privalo suprasti kaip kiekvienas iš šių karkasų veikia ir kokias savybes jie realizuoja (angl. supported features). Taip pat atsižvelgti į šiuos faktorius, prieš pradedant kurti, kadangi tai gali sutaupyti labai daug laiko ateityje [Yas12, p.13]:

- Kūrimo lengvumas naudojant šį karkasą
- Paprastumas išmokyti naudotis karkaso funkcionalumu (intuityvumas, dokumentacija, integruotų kūrėjo aplinkų (IDE) pagalba naudojant šį karkasą)
- Kūrimo kaina, naudojant šį karkasą (priklauso nuo jo paplitimo, žmonių mokančių dirbti šiuo karkasu, vyraujančio išsilavinimo, vietovės, amžiaus)
- Daugia-platforminis suderinamumas.

- Kūrėjo ir kitų naudotojų atsiliepimai ir patirtis naudojant šį karkasą.
- Karkaso funkcionalumas (pvz. ar turi 3d kūrimo integruotą funkcionalumą, ar leidžia prieti prie „Wi-Fi“ modulio telefone ir kt.)

Šiuo metu egzistuoja 3 pagrindinės karkasų aplinkos, kuriose galima kurti mobiliąsias aplikacijas [Yas12, 12-13 p.], tai:

Originalūs, kartu su OS išleidimu, programuotojams pateikiami **PĮ kūrėjo paketai** (Native SDK).

Pagr. privalumai: didžiausias našumas [žr. 3 pav.] ir galimybė maksimaliai prieti OS funkcijų, ir aparatinės įrangos valdymo - kameros, vibracijos funkcijos, gravitacijos sensoriaus, geografinės vietovės palaikymo (GPS ir kiti), Wi-Fi palaikymo, pirštų antspaudų skaitytuvo, daugybinio prisilietimo (angl. multi-touch) naudojimo, sisteminių failų skaitymo/rašymo leistinu saugumo lygiu, prieigos prie duomenų bazių (SQL Lite) ir kt. Taip pat suteikia prieigą prie galingo API, kas supaprastina kūrėjo darbą, bei leidžia sukurti kad ir pačią komplikuočiausią aplikaciją, naudojantis šiuo įrankiu.

Didžiausi trūkumai: kiekvienai OS – unikalūs SDK, ir jį išmokti užima laiko. Jeigu tai uždaro kodo OS, tai ir geriausi kūrėjo įrankiai, sukurti spec. šiai OS, greičiausiai gali būti ne pigūs, kuriuos dar ir privalo naujinti dažnai ir nepigiai.

Globaliosios naujos kartos interneto kalbos (HTML5, CSS3, JavaScript ir jos karkasai – jQuery ir kt.).

Pagr. privalumai: kūrėjo terpės pasaulietiškumas, ilgaamžiškumas, paprastumas (skirta visiems, dokumentuota išsamiai, pagalba dažnai nemokama ir greitai), pigumas (moka visi)

Didžiausi trūkumai: Ribota prieiga prie telefono aparatinės įrangos (angl. hardware)

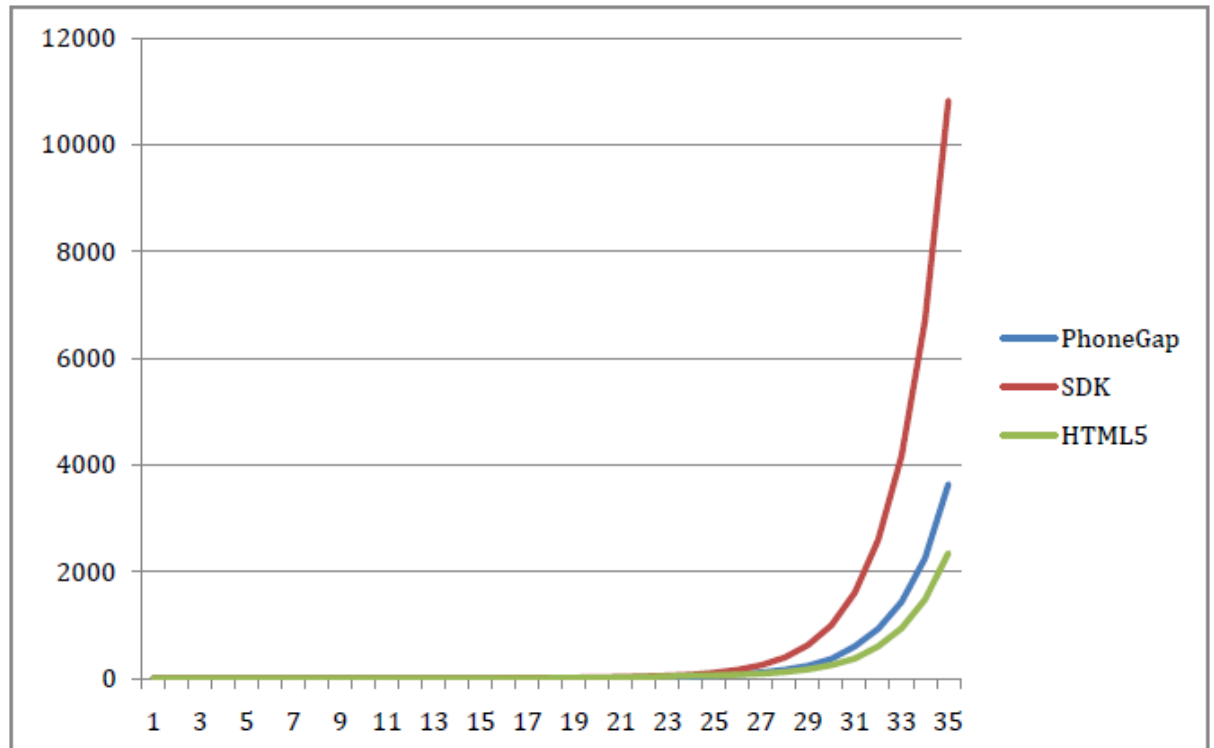
Ir 3-ųjų šalių sukurtos **daugia-platforminės PĮ kūrėjų aplikacijos** (PhoneGap (žr. 1 priedą), Appcelerator, MotherApp, ir kt.).

Pagr. privalumai: Daugiaplatformiškumas. Programuotojas gali kurti programinę įrangą sykiu daugeliui mobiliųjų OS, ir tai daryti tik viena, sau įprasta, programine kalba, nesigilindamas į kiekvienos iš OS specifiką. Taip pat turėti galimybę prieti prie daugelio, tik originaliam SDK prieinamų funkcijų, aparatūrinio spartinimo ir kt.

Didžiausi trūkumai: ne visada suspėja laiku su specifinės OS atnaujinimais, todėl naudojantis tokiu karkasu, gali nebūti galimybės pasinaudoti naujai išleista OS funkcija gan ilgą laiko tarpą.

1.8.1. Skirtingų karkasų aplinkų našumo palyginimas

Siekiant objektyviau nuspręsti karkaso aplinkos palyginimą, verta atsižvelgti į našumo testus vykdant algoritmus (žr. 3 pav.).



3 pav. Karkasų aplinkų našumo palyginimas (X - Fib(x), x reikšmė, Y - ms),
naudojant Android 2.3 OS paremtą HTC Sensation telefoną

2. MOBILIŲJŲ TELEFONŲ OS HIERARCHIJA IR APŽVALGA

2.1. Symbian, Maemo (MeeGo)

Kompanijos „Symbian Foundation“, kurią finansavo suomių kompanija „Nokia“, sukurta „Series 60 1th edition“ (dar vadinama „Symbian 6.1“) buvo pirmoji tikroji operacinė sistema, panaudota kompanijos „Nokia“ telefonuose, nors ji buvo labai riboto funkcionalumo.

Išleidus „Series 60 5th Edition“ (dar vadinama „Symbian 9.4“) operacinę sistemą, kuri galiausiai buvo priskirta prie Symbian pirmos kartos OS (Symbian^1), ši operacinė sistema jau buvo įgavusi labai didelį populiarumą išmaniųjų telefonų rinkoje ir užėmė dominuojančiąją padėtį.

Tačiau Symbian^1 (Series 60) iš esmės nebeatitiko šių dienų operacinių sistemų mobiliesiems telefonams poreikių, tokių kaip:

- Tiesioginis aplikacijų kompiliavimas į mobiliųjį telefoną, naudojant Nokia QT IDE.
- Daugybinius redaguojamus darbalaukius (angl. editable multi-desktop).
- Fono programas ir klientus (angl. background services, scheduled tasks).

PĮ kūrimas mobiliems telefonams buvo lėtas, nepatogus, neturėjo jokių naudingų greitų PĮ kūrimo galimybių, tai buvo tiesiog C++ programinio kodo, praplėsto tam tikru funkcionalumu (pvz. „Signals & Slots“ sistema), rašymas. (nebuvo sukurti tokie įrankiai kaip QT Quick ir QML). Todėl kompanija Nokia, norėdama sugebėti varžytis su kitų kompanijų mobiliųjų telefonų operacinėmis sistemomis turėjo atnaujinti Symbian operacinę sistemą, ką ir siekė padaryti sukurdamą Symbian^3 (arba Symbian 9.5, arba Series 60 v5.2) operacinę sistemą. Ji buvo išleista skubotai, todėl turi daugybę funkcijų, kurias reikėtų pataisyti, taip pat išlikusį senos kartos darbalaukį.

Po ne itin sėkmingo Symbian^3 starto ir reputacijos, kompanija Nokia paskelbė apie kartu su Intel kuriamą naujos kartos OS, pavadinimu Maemo, kurios pagrindu turėjo būti jau artimiausias, po Nokia N8 (pirmasis Symbian^3 mob. telefonas), pasirodysiantis telefonas. Tačiau praktiškai pabaigus Maemo, dėl smunkančių savo pozicijų rinkoje, 2010 m. pabaigoje ir 2011 pradžioje, „Nokia“ priėmė drastišką sprendimą ir jį įvykdė. Pirmiausia savo naujausiuose įrenginiuose pradėjo naudoti ne Maemo, bet Windows Phone 7 operacinę sistemą. Maemo pagrindu buvo sukurtas tik vienintelis telefonas – Nokia N9, o labai greitai po to buvo prezentuota ir šio telefono versija Windows Phone 7 OS pagrindu. Taip pat buvo pratęstas Symbian^3 OS (vėliau tapusios į Symbian Anna, o galiausiai Nokia Belle) sistemos palaikymas ir gyvavimas bent iki 2014 metų ir buvo nuspręsta kad 2011-aisiais ir dalį 2012-ųjų tai vis dar

bus pagrindinė kompanijos „Nokia” operacinė sistema. Tik 2012-ųjų antrame ketvirtyje pradėjo smarkiai daugėti Windows Phone 7 įrenginių, kurių didžioji dauguma planuojama išleisti 2012-ųjų antroje pusėje.

Be to, vietoje Symbian^3 OS greito šalinimo iš pardavimo ir vartojimo rinkos bei pakeitimo į Maemo operacinę sistemą, Maemo projektas buvo nukeltas į antrąjį planą, prijungtas prie kompanijos tyrimų (angl. Research) padalinio. Nutarta, kad galiausiai Maemo operacinė sistema bus prijungta prie dar vienos, kompanijos „Nokia” kuriamos MeeGo internetiniams-nešiojamiems kompiuteriams (Netbooks, TabletPC) skirtos operacinės sistemos, kuri turėtų būti konkurentė analogiškos paskirties kompanijos Google kuriamai operacinei sistemai „Chrome OS“.

Symbian^3 operacinė sistema 2011 metų viduryje ir 2012 metų pradžioje sulaukė rimtų patobulinimų ir atnaujinimų (Symbian Anna ir Nokia Belle versijos): buvo visiškai pakeista grafinė vartotojo sąsaja į atitinkančią šių dienų standartus ir galinčią varžytis su konkurentų mobilių OS grafinėmis vartotojo sąsajomis. Taip pat buvo ištaisyta daugybė šios OS turimų trūkumų (pvz., lange su vienu įvesties lauku, kur iki tol dar buvo būtina papildomai spustelti įvesties patį langą, norint pradėti vesti tekstą) ir nepatogumų (ribotas darbalaukio redagavimas, svarbios funkcijos sunkiau pasiekiamose OS vietose). Kadangi Maemo OS projektą buvo nuspręsta prijungti prie MeeGo OS, skirto internetiniams-nešiojamiems kompiuteriams, šiame darbe šios OS plačiau nenagrinėsime.

2.2. Google Android, Samsung Bada

Pirmasis Google Android pagrindu paremtas ir flagmanu tapęs gaminys buvo Samsung Galaxy S. Tai iš dalies įtakoją ir pakoregavo ir pačios kompanijos Samsung, kuri yra pagrindinė Android sistemų naudotoja šiuo metu, planus dėl savo operacinės sistemos pavadinimu „Bada OS”. Bada OS, pasak Samsung yra ir bus naudojamas funkcinuose telefonuose, tik žemos klasės išmaniuosiuose telefonuose [Kyl09], o Google Android – aukštos klasės išmaniuosiuose telefonuose. Pagrindinė priežastis, dėl kurios Samsung toliau plėtoja „Bada OS”, yra ta, jog Samsung yra išleidusi telefonų ir su „Windows Mobile” funkcinų telefonų OS ir siekia pakeisti šių telefonų būsimas versijas į naudojančias tik „Bada OS”.

Pačią Google Android operacinę sistemą galime suskirstyti į 4 kartų versijas:

1. Android 1.6 (Donut) – įtraukiant ir 1.0 – 1.5 versijas. Tai 1-os kartos, kompanijos Google sukurta OS, skirta išmaniesiems telefonams. Kadangi ši OS buvo gana ribota, turėjo nemažai trūkumų, o kūrėjo aplinka buvo labai sudėtinga (veikė tik kaip komandinė eilutė), trūko funkcionalumo (t.y. Ctrl+space nuspėjimo funkcija veikė labai ribotai, nebuvo tinkamos

Google App Inventor PĮ ir kt.), ši OS karta nesukėlė nei Android OS bumo nei revoliucijos išmaniųjų telefonų OS pramonėje.

2. Android 2.3 (Gingerbread) – įtraukiant ir 2.0 – 2.2 versijas, OS sistema skirta išmaniesiems mobiliems telefonams, o šios OS staigus išpopuliarėjimas ir bumas, privertė kompanijas Nokia ir Samsung atsisakyti planų apie naujų savų operacinių sistemų kūrimą ir ieškoti kitų sprendimų – Nokia perėjo prie Windows Phone 7, o Samsung savo „Bada” operacinę sistemą nutarė naudoti tik žemos klasės išmaniuosiuose telefonuose ir funkciniuose telefonuose. Kaip skelbia kompanija „Google”, 2011 m. pirmame ketvirtyje, kiekvieną mėnesį yra aktyvuojama po daugiau kaip 350 tūkst. Android 2.X įrenginių.
3. Android 3.1 (Honeycomb) – įtraukiant ir 3.0 versijas, OS sistema skirta didelės raiškos planšetiniams kompiuteriams. Šios OS versijos šiame darbe nenagrinėsime.
4. Android 4.0 (Ice Cream Sandwich) – versija, išleista 2011 metų pabaigoje, ir apjungianti Gingerbread funkcionalumą ir savybes, skirtas išmaniesiems telefonams, su „Honeycomb” planšetinių kompiuterių (angl. TabletPC), turinčiu 10 colių didelės raiškos ekraną, savybėmis [Sav11]. Taip pat ši operacinė sistema yra tinkama naudoti tiek išmaniuosiuose telefonuose, tiek planšetuose. Programinės įrangos kūrimas tokiu atveju nesiskiria.

Šiame darbe plačiau nagrinėsime būtent Android 2.3 (Gingerbread) OS, skirtą išmaniesiems telefonams ir jos funkcionalumą. Apie Google Android OS būtina paminėti dar vieną svarbų faktą – kompanija Google griežtai neigia faktą, kad savo Android OS naudoja Java kalbos branduolį ir Java virtualią mašiną (JVM) baitkodo interpretavimui. Tikėtina, kad būtent dėl finansinių ir autorinių priežasčių, yra teigiama, kad yra naudojama Dalvik vykdomąjį baitkodą (DEX - „Dalvik Executable”) interpretuojanti būtent DVM – Dalvik virtuali mašina (angl. „Dalvik Virtual Machine”), o ne JVM – Java virtuali mašina. Nors iš esmės tai yra ta pati Java kalba, su tomis pačiomis Java bibliotekomis. Tą liudija ir faktas, kad Android OS vienas pagrindinių kūrėjų yra žmogus, laikomas Java kalbos autoriumi ir šiuo metu dirbantis Google.

Bada OS taip pat galime suskirstyti į dviejų kartų versijas:

1. Bada 1.2 – įtraukia 1.0 – 1.2 – tai pirmoji Samsung išleistos operacinės sistemos versija, kol kas naudojama tik „Samsung Wave“ ir „Samsung Wave II“ vidutinės klasės išmaniuosiuose telefonuose. 2011 m. 1-ame ketvirtyje Samsung pardavė 3.5 milijono Bada OS paremtų telefonų, o tai visu 1 mln. daugiau nei įrenginių su WP7 OS pardavimų tuo laikotarpiu. Bada 1.X priskiriama prie funkcinių telefonų OS, nes neatitinka tikrosios OS apibrėžimo – nepalaiko „multi-tasking” funkcijos ne „native” tipo aplikacijose ir k.t.

2. Bada 2.0 – kol kas nėra įrenginių, naudojančių šią OS versiją. Pagaliau ši OS tapo modernia išmaniųjų telefonų OS, palaikančia daugiaprograminio režimo (angl. multi-tasking) funkcija ir kitas MOS apibrėžimą atitinkančias funkcijas.

Kadangi Bada 1.X nėra laikoma pilnaverte OS, o tokia esanti Bada 2.0 dar neturi ją aktyviai naudojančių įrenginių, tai šiame darbe išsamiai Bada OS nenagrinėsime. Tačiau plačiau apžvelgsime Bada 2.0 prie mobiliųjų OS kūrėjų perspektyvų.

3. IŠSAMI ESAMŲ IŠMANIŲJŲ TELEFONŲ MODERNIŲ OPERACINIŲ SISTEMŲ ANALIZĖ

3.1. Nokia Belle (S60 5.2, Symbian 10.1) operacinė sistema

3.1.1. Reikalavimai aparatinei įrangai (angl. hardware)

Sprendžiant iš bazinio telefono specifikaciją, norint pilnai naudotis Nokia Belle OS, reikalingas įrenginys su 600 Mhz ar greitesniu procesoriumi ir 256 MiB ar didesne operatyviaja atmintimi.

3.1.2. OS įvesties būdai, privalumai ir trūkumai

Palaikomi įvesties būdai:

- Klaviatūra;
- lietimui jautrus ekranas (angl. touchscreen).

Privalumai:

- „multi-touch“;
- palaikomi gamintojo ir 3-ųjų šalių foniniai servisai (angl. background services);
- „over the air“ PĮ atnaujinimai („OTASL“);
- OTASL atnaujinimai į kitą kartą (angl. major upgrades) – tai vienintelė OS (iš palyginamų), turinti šią funkciją tarp visų lyginamų;
- MIDP 2.1 Java aplikacijų kūrimo specifikacijų šablono palaikymas – vienintelė OS (iš palyginamų), palaikanti šią versiją;
- „cut“, „copy“, „paste“ funkcijų palaikymas;
- „*.XLS“, „*.DOC“ dokumentų redagavimas;
- GPS žemėlapių įdiegiami telefone, ir GPS sistema nereikalauja interneto ryšio;
- Nors pirminė versija, pavadinimu Symbian^3 (Symbian 9.5) buvo neišbaigta, vėliau sekę Anna ir Belle pataisymai, išsprendė daugybę problemų, ir padarė šią OS veikiančią gana sklandžiai;
- Nokia Belle pilnai palaiko HTML5/CSS 3.0 standartus. (ko nebuvo originalioje Symbian^3) [Fir11].

Trūkumai:

- QML karkasų nebuvimai;

- Šiuo metu vis dar esantis OS neišbaigtumas – bereikalingi paspaudimai, sulėtėjimai. Nors lyginant Symbian^3 ir Nokia Belle, situacija ir buvo žymiai pagerinta;
- Sudėtingas, ribotas, netikslus ir nepatogus derinimas (angl. debug);
- Nepalaikoma funkcija „atšaukti“ (angl. undo). Ji buvo numatyta tik „Maemo” platformai. Yra tikėtina, kad ši funkcija atsiras ateityje, kaip Symbian^3 atnaujinimas;
- Palaikoma tik „Lite“ Adobe Flash grotuvo versija.

3.1.3. Reikalavimai programuotojui ir būtina PĮ

Programuotojas turi mokėti dirbti Nokia QT4.8aplinkoje, būti susipažinęs su C++, Qt Quick ir QML galimybėmis bei Nokia Belle API klasių funkcionalumu. Taip pat naudotis Microsoft (C), „Windows XP/Vista/7” arba „Linux” operacine sistema.

Norint kurti aplikacijas Nokia Belle operacinei sistemai, būtina ši PĮ:

- GNU GCC ar pan. kompiliatorius
- Nokia QT 4.8
- Nokia BelleSDK

3.1.4. IDE ir API savybės programuotojui kuriant GUI sąsają

3-ųjų šalių PĮ kūrėjai gali kurti PĮ naudodami prieinamas Nokia Belle API klases, tačiau aplikacijos, naudojančios tam tikrą funkcionalumą turi būti pasirašytos skaitmeniu būdu (angl. digitally signed).

Grafinės vartotojo sąsajos kūrimas vyksta QT Quick IDE drag n’ drop įrankiu, taip pat QML deklaratyvia kalba.

GUI sąsajos kūrimo pavyzdžiai ir papildoma informacija pateikta 2 priede, prie šios OS.

3.1.5. Naudotojo sąsajos (UI) patogumas naudotojui

Naudotojo sąsajos (UI) vaizdai (žr. 4 pav.):



4 pav. Symbian^3 (9.5) vartotojo sąsajos vaizdai

3.1.6. Aplikacijų portabilumo galimybės

QML deklaratyvioje kalboje parašytos aplikacijos, taip pat aplikacijos, parašytos Python kalba ir skirtos Python interpretatoriui Symbian platformoje, padarius nedidelius pakeitimus, yra portabilios.

3.1.7. Trimatės grafikos ir žaidimų kūrimo galimybės

OpenGL ES bibliotekų palaikymas, aparatiškai spartinamas grafikos procesorius. Dėl C++ kalbos savybių, žaidimų kūrimas nėra paprastas, tačiau dėl GTK++ ir kitų bibliotekų kurti sudėtingus žaidimus šiai OS, nors ir sudėtinga, bet įmanoma.

3.1.8. API funkcionalumas programuotojui prieiti prie OS branduolio servisų

Programuotojas turi prieigą prie tam tikrų sisteminių servisų, pagal tai, kokį saugumo sertifikatą turi. Aukščiausią sertifikatą turi tik Nokia ir jos vidiniai PĮ kūrėjai bei partnerinės kompanijos, kuriančios sisteminės PĮ atnaujinimus ir pataisymus. 3-ųjų šalių kompanijos gali gauti prieigą prie:

- SMS žinučių serviso,
- GPS koordinačių serviso,
- magnetometro serviso,
- ir kitų ne branduolio (pvz. atnaujinimų) servisų.

3.2. Windows Phone 7

3.2.1. Reikalavimai aparatinei įrangai (angl. hardware)

Microsoft, kaip ir Google savo Android OS, bei ko nėra padarę likę išmaniųjų telefonų OS kūrėjai, yra paskelbusi sisteminių reikalavimų įrenginiui, kuris galėtų naudoti Windows Phone 7 operacinę sistemą, sąrašą (žr. 7 lent.):

7 lentelė. WP7 minimalūs reikalavimai

Minimalūs Windows Phone 7 įrenginio reikalavimai	
Ekranas	Bent 4-taškų „multi-touch“ tipo ekranas su bent WVGA (480x800) rezoliucija
CPU	800Mhz ar galingesnis ARM v7 "Cortex/Scorpion" tipo procesorius
GPU	DirectX9 palaikantis grafinis procesorius
RAM	256 MiB ar daugiau
Atmintis	8 GiB ar daugiau
Funkcijos	Akselerometras su kompasu, aplinkos apšviestumo jutiklis, artumo jutiklis, GPS gaviklis, giroskopas
Kamera	5MP kamera su LED blykste
Radijas	FM radijo tiuneris
Įvestis	7 atskiri fiziniai klavišai – atgal, pradėti, ieškoti, 2-o lygio kameros, įjungti/miegoti, garsinti ir tildyti.

3.2.2. OS įvesties būdai, privalumai ir trūkumai

Palaikomi įvesties būdai:

- klaviatūra;
- lietimui jautrūs ekranai (angl. touchscreen)

Privalumai:

- „multi-touch“;
- „over the air“ PĮ atnaujinimai („OTASL“);
- palaikomi gamintojo ir 3-ųjų šalių foniniai servisai (angl. background services);
- „cut“, „copy“, „paste“ funkcijų palaikymas (tik nuo 7.1 versijos);
- „Adobe Flash Player 10.1“ palaikymas (kitos OS palaiko tik „Flash Lite“ versijas);
- „*.XLS“, „*.DOC“ dokumentų redagavimas;
- pilnai palaikoma funkcija „atšaukti“ (angl. undo);
- „CTRL+Z“ klavišų kombinacija, iškviečianti šią funkciją.

Trūkumai:

- Licencijavimas – kūrėjai apriboti Windows OS kūrimo terpe, bene vienintelė rimta integruota kūrėjo aplinka (IDE) – Visual Studio, ir primygtinis Direct3D variklio naudojimas joje.
- Nebus OTASL atnaujinimo į „Windows Phone 8” versiją.
- GPS žemėlapiai nėra įdiegiami į telefoną, ir GPS sistema reikalauja interneto ryšio.

3.2.3. Reikalavimai programuotojui ir būtina PĮ

Programuotojas turi mokėti dirbti Visual Studio 2010 IDE aplinkoje, būti susipažinęs su C# .NET 4.0 galimybėmis bei Windows Phone 7 API klasių funkcionalumu. Taip pat naudotis © Microsoft „Windows Vista/7” operacine sistema.

Norint kurti aplikacijas Windows Phone 7 operacinei sistemai, būtina ši PĮ:

- Visual Studio 2010 IDE;
- Windows Phone Developer Tools RTW for Visual Studio 2010.

3.2.4. IDE ir API savybės programuotojui kuriant GUI sąsają

3-ųjų šalių PĮ kūrėjai gali kurti PĮ naudodami prieinamas Windows Phone 7 API klases, tačiau aplikacijos, naudojančios tam tikrą funkcionalumą turi būti pasirašytos skaitmeniu būdu (angl. digitally signed).

Grafinės vartotojo sąsajos kūrimas vyksta „Drag n’ drop” principu Visual Studio 2010 aplinkoje. GUI sąsajos kūrimo pavyzdžiai ir papildoma informacija pateikta 2 priede, prie šios OS.

3.2.5. Naudotojo sąsajos (UI) patogumas naudotojui

Naudotojo sąsajos (UI) vaizdai (žr. 5 pav.):



5 pav. Windows Phone 7 vartotojo sąsajos vaizdai

3.2.6. Aplikacijų portabilumo galimybės

C# .NET platforma yra portabili ir plačiai naudojama. Egzistuoja ir Python interpretatorius C# kalboje – „IronPython“. Pati .NET platforma naudojama tiek „(C) Microsoft Windows“ aplikacijoms kurti, tiek „ASP.NET“ interneto tinklapių kūrimui.

3.2.7. Trimatės grafikos ir žaidimų kūrimo galimybės

Windows, o kartu ir Windows OS yra itin orientuota į trimatę grafiką, dėl kompanijos Microsoft plėtojamo „DirectX“ grafinio variklio su „Direct3D“ ir kitų bibliotekų palaikymu. Taip pat pačių žaidimų kūrimui yra sukurta atskirta terpė „XNA Game Studio“, kurios realizacija taip pat yra ir Microsoft Visual Studio 2010 IDE. Tarp visų lyginamų OS, WP7 OS turi didžiausias, lengviausias, greičiausias ir patogias galimybes kurti itin sudėtingus trimačius žaidimus mobiliems telefonams.

3.2.8. API funkcionalumas programuotojui prieiti prie OS branduolio servisų

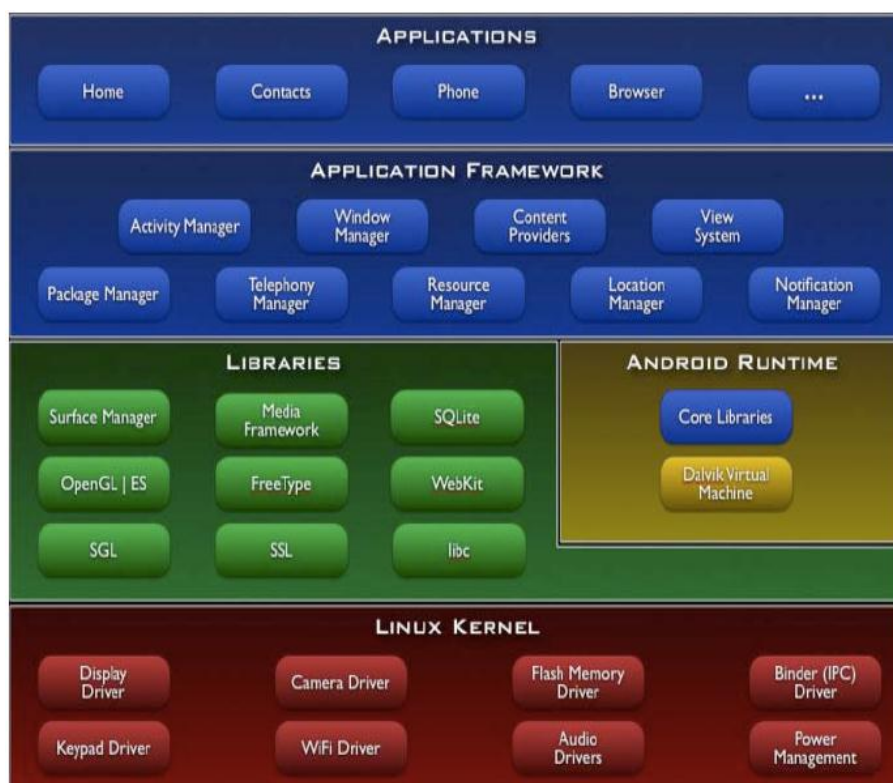
.NET 4.0 leidžia prieiti prie daugumos sisteminių funkcijų, tačiau yra daugybė saugumo sistemų, neleidžiančių tiesiogiai prieiti prie atminties (angl. direct memory access) saugaus programavimo režimu (priminsime, kad C# programavimo kalboje programinis kodas skirstomas į dvi dalis – „saugų“ („safe“ kodo blokas, įjungtas pagal nutylėjimą), ir „nesaugų“ („unsafe“ kodo blokas, leidžiamas tik specialių direktyvų pakeitimu).

3.3. Google Android 2.3

3.3.1. Architektūra, kompiliavimas, derinimas (angl. debug)

6 pav. matome pateiktą Android OS architektūrą. Kiekviena aplikacija Android OS turi savo procesą, kurių kiekvienas sukuria savo virtualią mašiną (VM). Kiekviena VM prisiskiria laisvą portą (angl. port), kad derinimo PĮ (angl. debugger) (DDMS serveris ar kita) galėtų per jį prisijungti prie programos [Yas12, 22 psl.].

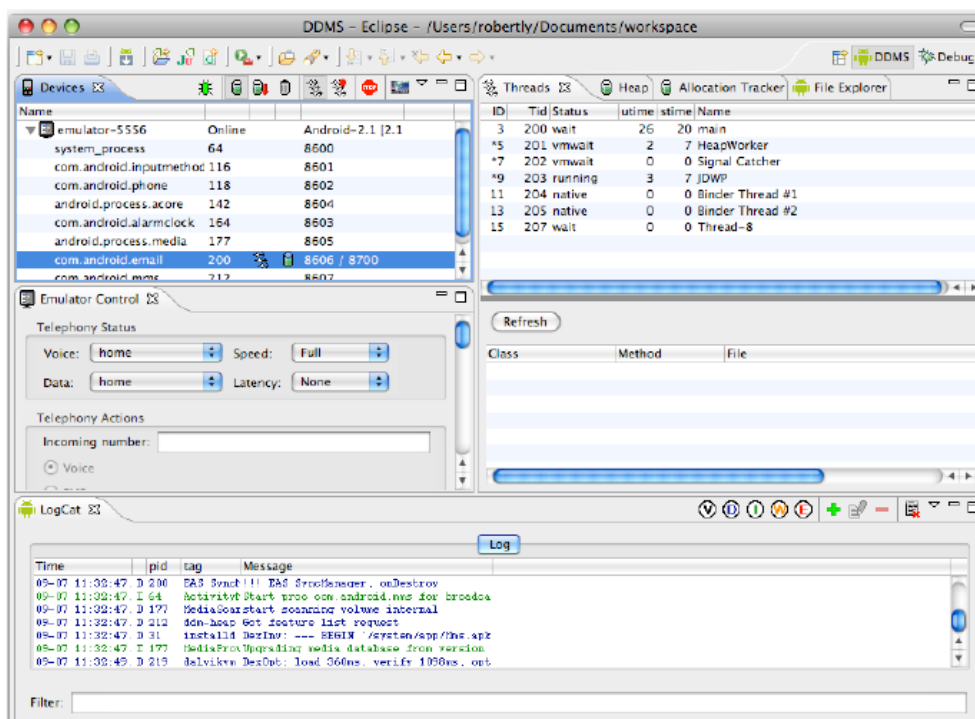
DDMS, vos tik pradėjęs darbą prisijungia prie Android derinimo tilto (angl. Android Debug Bridge (adb)) [Goo10b]. Tuomet yra sukuriamas VM stebėjimo servisas tarp adb ir DDMS kai tik įrenginys su Android OS yra prijungiamas. Šis servisas praneša DDMS serveriui, kada VM įrenginyje buvo paleista (angl. started) arba nutraukta (angl. terminated). Įprastai pirmas priskiriamas derinimo portas yra 8600, toliau 8601 ir t.t. Visas duomenų srautas yra nukreipiamas pirmiausia pereiti pro derinimo PĮ (angl. debugger).



6 pav. Android OS architektūra

Android OS derinimas vykdomas naudojant DDMS (Dalvik Debug Monitor Server, žr. 7 pav.) – tai derinimo įrankis, skirtas Android SDK, kuris suteikia portų-persiuntimo paslaugą (angl. port-forwarding services), gijų (angl. thread) ir steko viršūnės (angl. heap) duomenis įrenginyje, „logcat“, procesų sąrašą, įrenginio ekrano vaizdą, radijo būsenos informaciją,

vietovės duomenų paiešką naudojant GPS, įeinančių skambučių informaciją, SMS simuliaciją (angl. spoofing).



7 pav. DDMS derinimo įrankis naudojamas Eclipse aplikoje (IDE)

3.3.2. Reikalavimai aparatinei įrangai (angl. hardware)

Šiuo atžvilgiu kompanija Google pateikia gausų informacijos kiekį. Pagrindinis jų šaltinis – tai Android aplikacijų suderinamumo apibrėžimo dokumentas – CCD (angl. Android Compatibility Definition Document) ^{[Goo10],[Goo11c]}, kuriame itin išsamiai nurodomi reikalavimai telefono aparatui, kuriame gali veikti Android 2.3 OS. Pagrindiniai reikalavimai pateikti 8 lentelėje.

8 lentelė. Android 2.3 minimalūs reikalavimai

Minimalūs Google Android 2.3 įrenginio reikalavimai [Dkd10]	
Ekranas	Bent 2.5 colio istrižainė Bent 100 DPI taškų tankis Formatas tarp 4:3 ir 16:9
CPU	1Ghz ar galingesnis ARM v7 "Cortex/Scorpion" tipo procesorius
GPU	3D aparatinio būdų spartinamas („3d acceleration hardware“) grafinis procesorius, palaikantis bent OpenGL ES 1.0, taip pat gali būti reikalaujama ir OpenGL ES 2.0 palaikymo kai kuriais atvejais.
RAM	Bent 278 MiB: - 128 MiB ar daugiau branduoliui - 150 MiB ar daugiau aplikacijoms Kiekvienai paleidžiamai aplikacijai reikia: - Mažos raiškos ekrane: bent 16 MiB RAM - Didelės raiškos ekrane: bent 24 MiB RAM
Atmintis	Bent 2 GiB:

	<ul style="list-style-type: none"> - 1 GiB ar daugiau instaliacijoms, atnaujinams - 1 GiB ar daugiau – bendrai saugyklos
Jutikliai	PRIVALOMI: Akselerometras, magnetometras, giroskopas, GPS imtuvas, NEPRIVALOMI: barometras, aplinkos apšviestumo jutiklis, artumo jutiklis NEREKOMENDUOJAMI: termometras
Kamera	PRIVALOMA: 2MP NEPRIVALOMA: 2-oji priekinė kamera, blystė
Naršyklė	HTML5/W3C standartus atitinkančios naršyklės tvarkingas veikimas telefone
Ryšys	GSM arba CDMA, WiFi 802.11 b/g/n, Bluetooth
Dekoderiai	Aparatūriniai H.264 (.mp4/.3gp HD), JPEG ir kt. dekodieriai
Įvestis	Programinė („softkey“) arba fizinė („hardkey“) klaviatūra, atitinkanti bent vieną iš apibrėžtų klasėse: android.content.res.Configuration.keyboard (t.y. QWERTY klaviatūra arba 12-os klavišų) Taip pat privalo būti atskiri fiziniai „paieskos“, „namų“, „menu“, „atgal“ mygtukai, arba programiniai, tačiau pasiekiami visada.

Taip pat įrenginyje turi būti interpretatorius vykdamas „Dalvik Executable“ (DEX) baitkodą ir atitinkantis DVM (Dalvik Virtual Machine) specifikacijas ir semantiką.

Antrasis reikalavimus įrenginiui apibrėžiantis faktorius yra suderinamumo testų paketas – CTS (Compatibility Test Suite) [Goo11d]. Tik visus testus sėkmingai atlikęs mobilus įrenginys gali sėkmingai naudoti Android operacinę sistemą.

3.3.3. OS įvesties būdai, privalumai ir trūkumai

Palaikomi įvesties būdai:

- klaviatūra
- lietimui jautrūs ekranai (angl. touchscreen)

Privalumai:

- „multi-touch“
- „over the air“ PĮ atnaujinimai („OTASL“)
- „cut“, „copy“, „paste“ funkcijų palaikymas
- „*.XLS“, „*.DOC“ dokumentų redagavimas
- Itin draugiška sisteminių pranešimų (angl. notification) sistema [Sha11, 3 psl.]

Trūkumai:

- Dideli sisteminiai reikalavimai
- Palaikoma tik „Lite“ Adobe Flash grotovo versija.
- Nepalaikomi 3-ųjų šalių „fono servisai“ (angl. „3rd party background services“)
- Nėra OTASL atnaujinimo į „Android 4.X“
- GPS žemėlapių nėra įdiegiami į telefoną, ir GPS sistema reikalauja interneto ryšio
- Nepalaikoma funkcija „atšaukti“ (angl. undo).

3.3.4. Reikalavimai programuotojui ir būtina PĮ

Programuotojas turi mokėti dirbti Eclipse IDE aplinkoje, būti susipažinęs su Java 6 galimybėmis bei Android 2.3 API klasių funkcionalumu. Taip pat naudotis (C) Microsoft „Windows XP/Vista/7“ arba „Linux“ operacine sistema.

Norint kurti aplikacijas Android 2.3 operacinei sistemai, būtina ši PĮ [Ken10]:

- „JRE 6“
- „Eclipse IDE“
- „Android SDK“ su DVM virtualia mašina
- ADT („AndroidDevelopment Tools“) priedą Eclipse IDE [Goo10c]

ADT išplėčia Eclipse IDE funkcionalumą[Goo10c], todėl sukurti naują projektą tampa labai paprasta ir greita [Yas12, 22 psl.].

3.3.5. IDE ir API savybės programuotojui kuriant GUI sąsają

3-ųjų šalių PĮ kūrėjai gali kurti PĮ naudodami prieinamas Android 2.3 API klases, tačiau aplikacijos, naudojančios tam tikrą funkcionalumą turi būti pasirašytos skaitmeniu būdu (angl. digitally signed).

Grafinės vartotojo sąsajos kūrimas vyksta drag n' drop pagrindu „Google App Inventor“ IDE aplinkoje.

3.3.6. Naudotojo sąsajos (UI) patogumas naudotojui

Naudotojo sąsajos (UI) vaizdai (žr. 8 pav.):



8 pav. Android 2.3 vartotojo sąsajos vaizdai

3.3.7. Aplikacijų portabilumo galimybės

Iš esmės Android – tai Java kalba ir Java kalbos funkcionalumas. Todėl Android PĮ kodas gali būti su pakeitimais perkeliamas į Java SE 6 programų kodą, taip pat yra Python interpretacija Java kalbai (Jython), todėl aplikacijos su nedideliais pakeitimais yra portabilios.

3.3.8. Trimatės grafikos ir žaidimų kūrimo galimybės

Kiekvienas Android 2.3.įrenginys privalo turėti bent 1Ghz centrinį procesorių ir modernų aparatiniu būdu spartinamą grafinį procesorių (angl. accelerating hardware), palaikantį bent jau OpenGL ES 1.0 ir rekomenduojama, kad palaikytų ir OpenGL ES 2.0. Taip suteikiamos didelės galimybės kurti sudėtingos grafikos žaidimus mobiliajame telefone [Hin10].

3.3.9. API funkcionalumas programuotojui prieiti prie OS branduolio servisų

Android API yra itin platus ir leidžiantis pasiekti visus bazinius sisteminius servisus [Hin10]:

- SMS žinučių servisą
- Skambučių servisą
- GPS koordinačių servisą
- Kameros servisą
- Jungčių servisą
- Energijos valdymo servisą

3.4. Apple iOS 5.1

3.4.1. Reikalavimai aparatinei įrangai (angl. hardware)

Sprendžiant iš bazinio telefono specifikacijų, norint pilnai naudotis iPhone 5.1.X OS, reikalingas įrenginys su 1Ghz ar greitesniu procesoriumi, 512 MiB ar didesne operatyviaja atmintimi bei „PowerVR SGX535“ ar galingesniu grafiniu procesoriumi.

3.4.2. OS įvesties būdai, privalumai ir trūkumai

Palaikomi įvesties būdai:

- klaviatūra
- lietimui jautrus ekranai (angl. touchscreen)

Privalumai:

- „multi-touch“ (tik nuo iOS4)
- „cut“, „copy“, „paste“ funkcijų palaikymas

Trūkumai:

- „Adobe Flash“ nepalaikymas
- Nepalaikomi 3-ųjų šalių „fono servisi“ (angl. „3rd party background services“)
- Nepalaikomas „*.XLS“, „*.DOC“ dokumentų redagavimas, galimas tik jų skaitymas
- Nedraugiška pranešimų (angl. notification) sistema [Sha11, 3 psl.]
- Nepalaikomi „over the air“ PĮ atnaujinimai („OTASL“)
- GPS žemėlapiai nėra įdiegiami į telefoną, ir GPS sistema reikalauja interneto ryšio
- Nepalaikoma funkcija „atšaukti“ (angl. undo).

3.4.3. Reikalavimai programuotojui ir būtina PĮ

Programuotojas turi mokėti dirbti „XCode 4 developer tools“ IDE aplinkoje, būti susipažinęs su Objektinio C galimybėmis bei iOS 5.1 API klasių funkcionalumu. Taip pat naudotis (C) Apple „MacOS X“ operacine sistema.

Norint kurti aplikacijas iOS 5.1 operacinei sistemai, būtina ši PĮ:

- Objektinio C kompiliatorius
- iOS 5.1 SDK
- XCode 4 IDE [App12]

3.4.4. IDE ir API savybės programuotojui kuriant GUI sąsają

3-ųjų šalių PĮ kūrėjai gali kurti PĮ naudodami prieinamas iOS 5.1 API klases, tačiau aplikacijos, naudojančios tam tikrą funkcionalumą turi būti pasirašytos skaitmeniu būdu (angl. digitally signed). Grafinės vartotojo sąsajos kūrimas vyksta „XCode 4“ integruotos kūrimo aplinkos drag n' drop įrankiais.

3.4.5. Naudotojo sąsajos (UI) patogumas naudotojui

Naudotojo sąsajos (UI) vaizdai (žr. 9 pav.):



9 pav. iOS vartotojo sąsajos vaizdai

GUI sąsajos kūrimo pavyzdžiai ir papildoma informacija pateikta 2 priede, prie šios OS.

3.4.6. Aplikacijų portabilumo galimybės

Aplikacijos parašytos objektine C kalba, ir yra beveik neportabilios – yra sutapimai tik su pačių Apple naudojama MacOS X operacine sistema, skirta stacionariems kompiuteriams.

3.4.7. Trimatės grafikos ir žaidimų kūrimo galimybės

Visi iOS įrenginiai turi grafinį procesorių, bei gana spartų centrinį procesorių. Grafinis procesorius palaiko OpenGL ES bibliotekas, todėl trimačius žaidimus kurti nėra paprasta (trūksta tam skirtų taikomųjų programų ir patogių, greitų, trimačių žaidimų kūrėjo įrankių), bet yra įmanoma, nors Apple ir nėra kompanija, kurianti OS, orientuotas į žaidimus. [Mic10]

3.4.8. API funkcionalumas programuotojui prieiti prie OS branduolio servisu

Dėl sistemos saugumo nustatymų, programuotojams neleidžiama prieiti prie šių sisteminių servisu: SMS žinučių serviso ir skambučių serviso.

3.5. HP/Palm webOS 2.2

3.5.1. Reikalavimai aparatinei įrangai (angl. hardware)

Sprendžiant iš bazinio telefono specifikacijų, norint pilnai naudotis webOS, reikalingas įrenginys su 800 Mhz ar greitesniu procesoriumi, 512 MiB ar didesne operatyviaja atmintimi ir OpenGL ir Direct3D mobile bibliotekas palaikančiu grafikos procesoriumi.

3.5.2. OS įvesties būdai, privalumai ir trūkumai

Palaikomi įvesties būdai:

- Klaviatūra
- Lietimui jautrūs ekranai (angl. touchscreen)
- Privalumai:
- „multi-touch“
- „over the air“ PĮ atnaujinimai („OTASL“)
- „cut“, „copy“, „paste“ funkcijų palaikymas

Trūkumai:

- „Backwards incompatible“ - webOS 2.2 nepalaiko PalmOS aplikacijų
- Palaikoma tik „Lite“ Adobe Flash grotuvo versija.
- Nepalaikomi 3-ųjų šalių „fono servisai“ (angl. „3rd party background services“)
- Nepalaikomas „*.XLS“, „*.DOC“ dokumentų redagavimas, galimas tik jų skaitymas
- GPS žemėlapiai nėra įdiegiami į telefoną, ir GPS sistema reikalauja interneto ryšio
- Nebus OTASL atnaujinimo į webOS 3.0 versiją
- Nepalaikoma funkcija „atšaukti“ (angl. undo).

3.5.3. Reikalavimai programuotojui ir būtina PĮ

Programuotojas turi mokėti dirbti Eclipse IDE aplinkoje, būti susipažinęs su C kalbos galimybėmis bei webOS 2.2 API klasių funkcionalumu. Taip pat naudotis (C) Microsoft „Windows XP/Vista/7“ arba „Linux“ pobūdžio operacine sistema.

Norint kurti aplikacijas webOS2.2 operacinei sistemai, būtina ši PĮ:

- JRE 6 (Eclipse) ir GNU GCC ar pan. Interpretatoriaus;
- Eclipse IDE;
- webOS developer plugin for Eclipse.

3.5.4. IDE ir API savybės programuotojui kuriant GUI sąsają

3-ųjų šalių PĮ kūrėjai gali kurti PĮ naudodami prieinamas webOS 2.2 API klases, tačiau aplikacijos, naudojančios tam tikrą funkcionalumą turi būti pasirašytos skaitmeniu būdu (angl. digitally signed).

Grafinės vartotojo sąsajos kūrimas vyksta naudojant Mojo kūrimo karkasą [Mil09]:

- Bendri UI elementai
- HTML ir Javascript kodo rašymas GUI kūrime
- JSON bibliotekų palaikymas
- MVC – modelių/vaizdų valdiklis (angl. Model View controller)
- Drag n' drop
- GUI sąsajos kūrimo pavyzdžiai ir papildoma informacija pateikta 2 priede, prie šios OS.

3.5.5. Naudotojo sąsajos (UI) patogumas naudotojui

Naudotojo sąsajos (UI) vaizdai (žr. 10 pav.):



10 pav. webOS 2.2 vartotojo sąsaja

3.5.6. Aplikacijų portabilumo galimybės

Eclipse IDE ir C kalba yra vieninteliai portabilumo faktoriai. Python interpretatorius ar QML palaikymas šiai OS kol kas neegzistuoja. Todėl aplikacijos šioje OS yra neportabilios.

3.5.7. Trimatės grafikos ir žaidimų kūrimo galimybės

Šių faktorių labiausiai įtakoja grafikos procesorius, naudojamas išmaniajame telefone, t.y. pvz. „HP Veer“ naudojamas „Adreno 205“ grafikos procesorius (GPU) palaiko tokias technologijas:

- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenVG 1.1
- EGL 1.3
- Direct3D Mobile
- SVGT 1.2
- Direct Draw

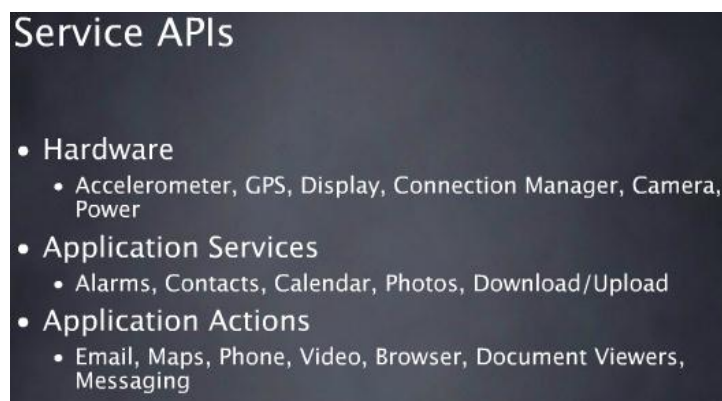
Taip pat: Posix, SLD, PDL. Kūrimas vyksta „Eclipse” IDE terpėje, rašant C kalbos kodą.

3.5.8. API funkcionalumas programuotojui prieiti prie OS branduolio servisų

Dėl C kalbos primityvumo, programuotojams leidžiama prieiti tiek prie dinaminio atminties valdymo, tiek prie branduolio bazinių servisų [DM10]:

- SMS žinučių serviso,
- skambučių serviso,
- GPS serviso,
- kameros serviso,
- jungčių valdiklio,
- energijos valdymo.

Išsamesnę informaciją matome 11-ame paveikslėlyje.



11 pav. webOS API servisų prieiga

3.6. Research in motion (RIM) Blackberry OS 7.1

3.6.1. Reikalavimai aparatinei įrangai (angl. hardware)

Sprendžiant iš bazinio telefono specifikacijų, norint pilnai naudotis Blackberry OS, reikalingas įrenginys su 600 Mhz ar greitesniu procesoriumi ir 256 MiB ar didesne operatyviaja atmintimi.

3.6.2. OS privalumai ir trūkumai

Palaikomi įvesties būdai:

- Klaviatūra;
- „trackwheel“;
- „trackball“;
- „trackpad“;
- Lietimui jautrūs ekranai („touchscreen“).

Privalumai [McI10]:

- „multi-touch“ (tik nuo 6.0);
- „over the air“ PĮ atnaujinimai („OTASL“);
- „cut“, „copy“, „paste“ funkcijų palaikymas;
- „*.XLS“, „*.DOC“ dokumentų redagavimas;
- Ribotai palaikoma funkcija „atšaukti“ (angl. undo).

Trūkumai [McI10]:

- Nėra galimybė atnaujinti Blackberry 6.0 įrenginių į 7.0 versiją;
- Palaikoma tik „Lite“ Adobe Flash grotuvo versija;
- Nepalaikomi 3-ųjų šalių „fono servisai“ (angl. 3rd party background services);
- GPS žemėlapiai nėra įdiegiami į telefoną;
- ir GPS sistema reikalauja interneto ryšio.

3.6.3. Reikalavimai programuotojui ir būtina PĮ

Programuotojas turi mokėti dirbti Eclipse IDE aplinkoje, būti susipažinęs su Java 6 galimybėmis bei Blackberry 7 API klasių funkcionalumu. Taip pat naudotis (C) Microsoft Windows XP/Vista/7 arba (C) Apple MacOS X operacine sistema.

Norint kurti aplikacijas Blackberry 7 operacinei sistemai, būtina ši PĮ:

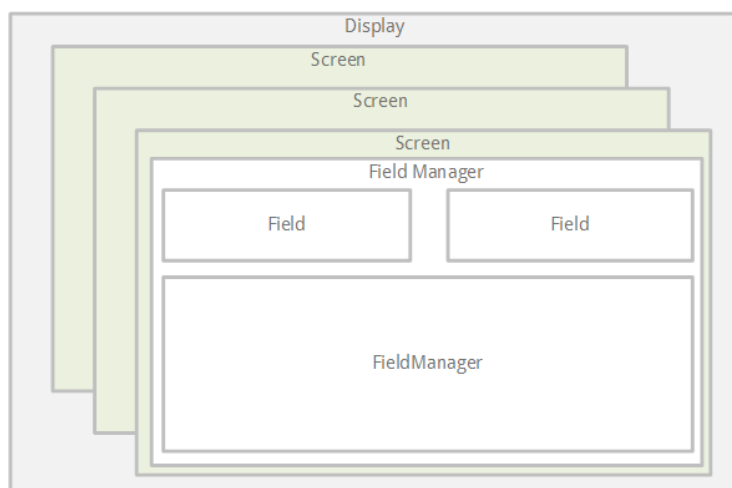
- JRE 6
- Eclipse IDE
- BlackBerry Java Plugin for Eclipse v1.3 MR

3.6.4. IDE ir API savybės programuotojui kuriant GUI sąsają

3-ųjų šalių PĮ kūrėjai gali kurti PĮ naudodami prieinamas Blackberry API klases, tačiau aplikacijos, naudojančios tam tikrą funkcionalumą turi būti pasirašytos skaitmeniu būdu (angl. digitally signed) [Naz09].

Grafinės vartotojo sąsajos kūrimas vyksta RIM UI API arba J2ME MIDP specifikacijos pagrindu, tačiau neegzistuoja joks „drag n drop“ GUI kūrimo įrankis. Pirmasis toks įrankis turėtų būti „Dragon RAD“ („Rapid Drag-And-Drop“) GUI programa, tačiau ji vis dar kuriama ir ji turėtų būti ne tik į Blackberry orientuota programa, bet ir daugiaplatforminė PĮ (IDE), standartizuojanti GUI kūrimą Blackberry, Android, Windows Phone platformose (angl. „a cross-platform RAD GUI development“). [Wil11, McI10]

Pats GUI kūrimas yra paremtas tokia struktūrine hierarchija (žr. 12 pav.):



12 pav. Blackberry GUI hierarchija

GUI sąsajos kūrimo pavyzdžiai ir papildoma informacija pateikta 2 priede, prie šios OS.

3.6.5. Naudotojo sąsajos (UI) patogumas naudotojui

Naudotojo sąsajos (UI) vaizdai (žr. 13 pav.):



13 pav. BlackBerry 6.0 vartotojo sąsaja

3.6.6. Aplikacijų portabilumo galimybės

Blackberry aplikacijos yra rašomos Java platformos pagrindu, todėl turi panašumų su Android, tačiau apskritai aplikacijos neportabilios ir tinkamos tik Blackberry platformoje. GUI sąsaja, jeigu parašyta pagal J2ME MIDP specifikaciją, turi nemažai panašumų su Nokia S40 aplikacijų programiniu kodu, kuris taip pat parašytas su Java. [Gos08]

3.6.7. Trimatės grafikos ir žaidimų kūrimo galimybės

Blackberry – orientuota į verslo klasę OS, t.y. į žmones, kurie mažai domisi žaidimais. 3-iųjų šalių kūrėjams pirmieji įrankiai buvo pateikti tik 2009 metais, ir jie yra labai riboti žaidimų kūrime, lyginant su kitomis išmaniųjų telefonų OS. Žaidimai kuriami Java programavimo kalba, pagal JDK 6 turimą funkcionalumą. [Ham10, Wes08]

Taip pat:

- nuo BlackBerry 4.6 pagal nutylėjimą įtrauktas SVG palaikymas,
- nuo BlackBerry 5.0 yra palaikomi OpenGL ES grafinės bibliotekos,
- nuo BlackBerry 6.0 yra palaikomos OpenVG bibliotekos.

3.6.8. API funkcionalumas programuotojui prieiti prie OS branduolio servisų

Skaitmeninių būdu patvirtintose programose, programuotojams leidžiama prieiti prie šių bazinių sistemos funkcijų:

- GPS koordinačių
- SMS žinučių serviso
- Skambučių serviso
- Kameros
- Įvesties/išvesties jungčių.

4. IŠMANIŲJŲ TELEFONŲ OS PALYGINIMAS

4.1. Išmaniųjų įrenginių aparatinės įrangos palyginimas

Pagal 9 lentelę galime palyginti standartines išmaniųjų telefonų komplektacijas pagal jų operacines sistemas:

9 lentelė. Baziniai išmanieji telefonai pagal OS

OS	Symbian^3	iOS 4.3	Android 2.3	Blackberry 6.0	Windows Phone 7	webOS 2
Įrenginys	Nokia N8	iPhone 4G [Gsm10]	Samsung Galaxy S	Blackberry 9300 Curve 3G	HTC HD7	HP Veer
Išleidimo Data (įrenginio)	2010-10	2010-06	2010-06	2011-02	2010-10	2011-05
CPU	680 MHz (ARM 11)	1 GHz (ARM Cortex-A8)		624 Mhz	1 GHz Scorpion	800MhzScorpion (ARM 7)
GPU	Broadcom BCM2727	PowerVR SGX535	PowerVR SGX540	---	Adreno 200	Adreno 205
RAM	256 MiB	512 MiB	512 MiB	256 MiB	576 MiB	512 MiB
ROM	512 MiB	---	2 GiB	---	512 MiB	---
Atmintis (Vidinė+SD)	16+32 GiB	16+0 GiB	8+32GiB	256 MiB+32GiB	8+32 GiB	8+0 GiB

4.2. Išmaniųjų telefonų su moderniomis OS funkcionalumo palyginimas

Remiantis įvairiais interneto šaltiniais, pateikiame apibendrinta įvairių išmaniųjų telefonų OS funkcionalumo lentelę (žr. 10 lentelę).

10 lentelė. Išmaniųjų OS funkcionalumas

Funkcija	Symbian10.1 (Nokia Belle)	iOS 5.1	Android 2.3	Blackberry 6.0	Windows Phone 7	webOS 2.2
Multi-tasking	Taip	Taip	Taip	Taip	Taip	Taip
Atnaujinimai (On-the-air / kitą kartą)	Taip / Taip	Ne/Ne	Taip / Ne	Taip / Ne (7.0)	Taip / Ne (WP 8)	Taip / Ne
Multi-touch	Taip	Taip	Taip	Taip (tik 6.0+)	Taip	Taip
Neorientuotas ekranas („Pivot“)	Taip					
„Cut“, „Copy“, „Paste“ (kirpti, kopijuoti, klijuoti)	Taip	Taip	Taip	Taip	Taip (tik 7.1+)	Taip
„Undo“ (atšaukti)	Ne	Taip	Ne	Ne	Taip	Ribota
Wi-Fi	802.11 b/g/n					
WAP	2.0					
HTML	5	5	5	5	5	5
CSS	3.0	3.0	3.0	3.0	3.0	3.0
Javascript	Taip	Taip	Taip	Taip	Taip	Taip
Ajax	Taip	Taip	Taip	Taip	Taip	Taip
jQuery	Nėra duom.	Taip	Taip	Nėra duom.	Taip	Taip
JSON	Nėra duom.	Taip	Taip	Nėra duom.	Nėra duom.	Taip
Java	MIDP 2.1	Ne	MIPD EMU	MIDP 2.0	JBED	MIDP 2.0
Flash	Lite v4.0	Ne	Lite v4.0	Lite v4.0	Player v10.1	Lite v4.0
PDF, DOC skait.	Taip/Taip	Taip/Taip	Taip/Taip	Taip/Taip	Taip/Taip	Taip/Taip
XLS, DOC redag.	Taip/Taip	Ne/Ne	Taip/Taip	Taip/Taip	Taip/Taip	Ne/Ne
Web apps	Nėra duom.	Taip	Taip	Nėra duom.	Nėra duom.	Taip

4.3. Mobiliosios OS PĮ kūrimo galimybės įvairiose OS su IDE

11-oje lentelėje matome, kokias PĮ išmaniųjų telefonų operacinėms sistemoms turi įvairių OS naudotojai-programuotojai.

11 lentelė. Personalinių kompiuterių ir išmaniųjų operacinių sistemų
kūrimo aplinkų suderinamumas

OS	Windows 7	Ubuntu 10	MacOS X	FreeBSD 8
Platforma	Windows NT	Linux	Unix	BSD
Symbian10.1 (Nokia Belle)	+ (QT)	+ (QT)	+ (QT)	+ (QT)
iOS 5.1.X	-	-	+ (Xcode 4)	-
Android 2.3	+ (Eclipse)	+ (Eclipse)	-	-
Blackberry 6.0	+ (Eclipse)	-	+ (Eclipse)	-
Windows Phone 7	+ (VS 2010)	-	-	-
webOS 2	+ (Eclipse)	+ (Eclipse)	-	-

4.4. Bendras išmaniųjų OS įvertinimas (1-10)

Išbandžius įvairius kūrėjo paketus (SDK) ir jiems skirtas integruotas kūrimo aplinkas (IDE), rašydamas bendrą įvertinimą, autorius, remdamasis SDK dokumentacija palygino kiekvienos išmaniosios OS pateikiamą API funkcionalumą (galimybes prieti pvz. prie Wi-Fi ryšio valdymo), išmanumą ir greitumą (ar OS pridedamas karkasinis funkcionalumas prie programavimo kalbos, pagreitina darbą ja, prideda naujų metodų, dažnai naudojamų, ir leidžiančių išvengti tokio kodo rašymo pačiam), taip pat remiantis informacija rasta įvairiuose interneto šaltiniuose, autorius įvertino įvairių išmaniųjų telefonų OS, jų kūrėjo aplinkas (IDE), žaidimų kūrimo galimybes, patogumą ir našumą programuotojui, dokumentacijos aiškumą ir parengė jų įvertinimo lentelę (žr. 12 lentelę).

12 lentelė. Bendras išmaniųjų OS įvertinimas

OS	Symbian	iOS	Android	Blackberry	Windows Phone 7	webOS
Versija	10.1	4.3	2.3.1	6.0.0	7.0.X	2.X
Kalba	C++	Objekt. C	Java 1.7	Java 1.7	C# .NET 4.0, XNA	C
Pagr. IDE	Qt 4.7	Xcode 4	Eclipse	Eclipse	Visual Studio 2010	Eclipse
Bendrai įvertinimas	8	7	6	5	9	6
PĮ kūrimo greitumas (kodas+ intuityvumas)	7	8	6	6	9	6
Kodo rašymo greitis	5	4	7	7	9	3
IDE prieinamumas	8	2	7	7	2	7
IDE patogumas	6	9	5	5	8	5
GUI kūrimo lengvumas	8 (QML)	7	6	2	10	8 (Mojo)
3D efektų kūrimo galimybės	10 (QML)	6	5	3	9 (SilverLight)	8
3D žaidimų kūrimo galimybės	8 (OpenGL ES)	-	8 (OpenGLES)	4 (OpenGL ES)	10 (XNA)	7 (OpenGLE S)
Derinimas (angl. Debug)	6	-	8	-	10	6
Dokumentacija	7	6	7	5	10 (MSDN)	6

4.4.1. Mobiliųjų OS palyginimo vertinimo kriterijai

12 lentelės (bendras išmaniųjų OS įvertinimas) įvertinimai buvo nustatyti šio bakalauro darbo autoriaus. Įvertinimai rėmėsi:

1. Asmenine patirtimi, bandant įvairias IDE, skirtas šioms OS. Kaip greitai pavyko sukurti pirmąją „Hello world“ programą, koks buvo dokumentacijos detalumas ir kaip lengvai dokumentacija buvo pasiekama internete.
2. Kiekvienos iš OS API galimybėmis/ribotumu.
3. Programine kalba, kuria yra parašyta ta OS, ir kuria, be karkaso suteikiamo papildomo funkcionalumo, yra kuriamos taikomosios aplikacijos įrenginiams su šia OS. Taip yra aktualu tokias atvejais, kaip pvz. „šiukšlių rinktuvas“ (angl. garbage-collector) – Symbian platformai kuriant aplikaciją Nokia QT, karkasas prideda . „šiukšlių rinktuvo“ funkcionalumą, tačiau jo nėra originalioje C++ kalboje, todėl dažnu atveju būtinas tokio funkcionalumo papildomas inicializavimas (kviesti klases su specifiniu parametru, įtraukti papildomas bibliotekas), bei jis yra kur kas labiau ribotas, mažiau veiksmingas (neįtrauktas į pačios kalbos branduolio funkcijų kompiliavimo procesą) ir tikėtina, turi daugiau klaidų/netikslumų nei tas pats, pvz. „šiukšlių rinktuvas“ realizuotas pačios kalbos baziniame lygmenyje.
4. Kitais panašaus pobūdžio straipsniais, kur yra įvertinamas įvairių karkasų našumas [Yas12], įvairių karkasų funkcionalumas [LY09, TSL12], platformos atviro kodo pranašumai ir trūkumai [HBH12].
5. Dviejų OS palyginimais – Android ir iOS [Che10,Sha11], Windows Phone ir Symbian [SGE09], ar (ir kaip dažnai) ši OS yra įtraukiama į multi-platforminių įrankių, skirtų kurti aplikacijas iškart keliomis OS, sąrašus [OT10].

Įvertinęs informaciją, gautą iš šių 1-5 punktų, autorius parašė įvertinimus nuo 1 iki 10, bei „-“ ten, kur nepavyko rasti patikimos informacijos. 10 – maksimalus įvertinimas, reiškiantis kad jokių trūkumų ar nesklandumų nebuvo pastebėta. 5 – minimalus įvertinimas, kuomet ta savybė naudotis įmanoma, nors ir su dideliais trūkumais, ir problemomis. Jeigu įvertinimas žemiau 5 – tos savybės suteikiama nauda neatperka nepatogumo ir neaiškumo, skiriamo bandant naudotis ta savybe.

5. ATEITIES PERSPEKTYVOS IR TENDENCIJOS

5.1. Išmaniųjų OS abstrakcijos ir globalumo tendencijos

Visos mobiliosios operacinės juda vis didesnės abstrakcijos ir globalumo link, t.y. po truputi nyksta riba tarp įvairių įrenginių – išmaniųjų telefonų, delninių kompiuterių, planšetinių kompiuterių ir internetinių nešiojamųjų kompiuterių.

Teigdamas šį teiginį, autorius remiasi šiais faktais:

Kompanijos Nokia - Maemo OS prijungimu prie MeeGo planšetinių/internetinių kompiuterių.

Kompanijos Google – išmaniųjų telefonų OS - Android 3.0/3.1 pritaikymu planšetiniams kompiuteriams, ir kompanijos Google planu metų bėgyje išleisti Google Android 4.0 versija, kuri turėtų apjungti išmaniųjų ir planšetinių kompiuterių šios OS versijas, bei matyt po šio žingsnio vėliau būsiančio internetiniams kompiuteriams (angl. netbooks) skirtos OS Google Chrome OS ir Google Android OS bendrinimo ir suvienodinimo, taip pat tai, jog Android OS palaiko ne tik ARM, t.y. mobiliųjų telefonų, bet x86 serijos procesorius, t.y. naudojamus ir stacionariuose kompiuteriuose, yra labai tikėtinas faktas kad kelių metų bėgyje kompanija Google pristatys vienu ar kitu pavidalų šią ar šias OS tinkančias tiek stacionariam kompiuteriui, tiek išmaniajam telefonui.

Kompanijos „Microsoft“ – iki šiol tik x86 (taip pat ir specifiniuose serverių architektūros procesoriuose, pvz. IA-64 ir kt.), planai į Windows 8 ir Windows 9 operacines sistemas jau pilnai įtraukti ir ARM procesorių palaikymą, taip pat Windows CE ir Windows NT platformų bendrinimo, kas galiausiai gali privesti prie vienos bendros sistemos. Taip pat Windows Mobile serijos OS nutraukimo ir perėjimo prie Windows Phone serijos OS, kas iš tiesų yra ta pati Windows NT serija, tik su itin sumažintu ir apribotu funkcionalumu ir poreikiais aparatūrinei įrangai.

Kompanijos „Apple“ – požiūriu ir reprezentavimu savo iOS operacinės sistemos, kaip tiesiog „Mac OS X“, skirtos įprastiems kompiuteriams („Desktop PC“ ir „Laptop“) realizacijos išmaniajame telefone, kaip įterpiamosios PĮ (angl. embedded software). Taip pat remiantis tuo faktu kad daugiau kaip metus laiko kompanija Apple iOS operacinei sistemai apskritai nebuvo pateikusi jokio kūrėjo paketo (SDK) ar programinės įrangos kūrimo sąsajos (API), motyvuodama tuo, kad „iOS“ operacinė sistema geba puikiai išnaudoti interneto aplikacijas (angl. web apps) telefone kaip taikomąsias programėles ir telefono funkcionalumo išplėtimą. Tačiau, šis sprendimas, matyt, buvo priimtas kiek per anksti pasaulinės rinkos atžvilgiu, kadangi

po daugiau nei metų laikotarpio nuo iPhone OS 1.0 pasirodymo, tokie SDK ir API visgi buvo sukurti, šiuo metu yra plačiai naudojami bei yra viena iš pagrindinių Apple iPhone įrenginio kulto, bei jo aplikacijų parduotuvės/parsisiuntimo „AppStore“ didelio populiarumo ir sėkmės priežasčių.

5.2. Debesų kompiuterijos ir interneto aplikacijų įtaka išmaniųjų telefonų OS

Debesų kompiuterija (angl. cloud-computing) ir interneto aplikacijos (angl. web apps) išmaniuosiuose telefonuose užims svarbią dalį. Šį faktą įtakos šie veiksniai:

- didelis paplitimas įrenginių, palaikančių bevielį interneto ryšį,
- bevielio interneto ryšio ilgalaikis pigimo procesas,
- bevielio duomenų perdavimo spartos ilgalaikis didėjimas,
- bevielio interneto prieigos galimybė praktiškai bet kurioje gyvenamoje pasaulio vietoje jau per artimiausius keletą metų,
- didelės investicijos ir bandymai spręsti „neišnaudojamų resursų“ problemą, t.y. dauguma vartotojų vidutiniškai išnaudoja apie 2-15 proc. savo turimų resursų, į kuriuos investavo (pvz. galingą aparatinę įrangą turintis kompiuteris, ar dviejų branduolių procesoriumi paremtas išmanusis telefonas).

Todėl laikui bėgant vis didesnė taikomųjų programų dalis bus perkeliama į nuotolinius-išorinius serverius (internetu tinklapius ir jų įskiepius). Tai savo ruožtu leis mažinti aparatūrinius reikalavimus pačiam prietaisui – mobilaus telefono aparatui su operacine sistema. Visa tai leis šiuos įrenginius (išmaniuosius telefonus) žymiai atpiginti, jie taps prienami kiekvienam norinčiajam šį įrenginį turėti.

5.3. Išmaniųjų telefonų OS ateities perspektyvos

Remiantis bakalaurinio darbo metu autoriaus sužinotą informaciją, OS kūrėjų kompanijų paskelbtais duomenimis ir įvairių statistiniais duomenimis užsiimančių kompanijų prognozėmis, galime padaryti tokias prielaidas:

- Visos mobiliosios OS juda vis didesnės abstrakcijos ir globalumo link, t.y. po truputi nyksta riba tarp įvairių įrenginių – išmaniųjų telefonų, delninių kompiuterių, planšetinių kompiuterių ir internetinių nešiojamųjų kompiuterių. (Šis teiginys yra pagrindžiamas šio darbo 5.1. poskyryje).

- Tikėtina, kad 10 metų laikotarpyje jau bus paruoštos tokios operacinės sistemos, kurios tiks tiek išmaniajam telefonui, tiek stacionariam kompiuteriui, tiek muzikos grotuvui, tiek e-knygai (žr. 5.1. poskyrį).
- Ilgalaikiu vis mažesnė aplikacijų dalis bus pačiuose telefonuose (žr. 5.2. poskyrį).
- Aparatūriniai reikalavimai išmaniesiems telefonams ilgainiui mažės, o tai leis šiuos įrenginius itin atpiginti, jie taps prienami kiekvienam (žr. 5.2. poskyrį).

5.4. Išvados ir teiginiai iš išmaniųjų OS rinkos prognozių

Remiantis 5.3. poskyryje pateiktomis išmaniųjų OS prognozėmis, galime teigti, jog:

1. dėl smarkiai išaugusiančio išmaniųjų telefonų kiekio,
2. dėl kartu smarkiai išaugusiančio ir operacinių sistemų rinkos dydžio,
3. parašius aplikaciją išmaniam telefonui, ji galės būti naudojama ir planšetiniame ir internetiniame ar net stacionariame kompiuteryje,
4. paties originalaus (C++, objektinio C, C#) programinio kodo rašymas yra paslepiamas po įvairiais karkasais (angl. frameworks), tokiais kaip QML, SilverLight, ar net pačių programavimo kalbų (pvz. Python) [Gos08] interpretavimo daugelio kalbų kompiliatoriais (pvz. C++ - CPython, Java - Jython, C# - IronPython ir kt.),
5. net ir šiuo metu, bene visą grafinę aplinką ir jos interaktyvumą galima sukurti iš vis neberašant programinio kodo, o tiesiog naudojantis integruotų kūrimo aplinkų („IDE“) suteikiamu „Drag ‘n drop“, tiesioginės peržiūros be kompiliavimo (pvz. „instant-preview“ funkcija QML deklaratyvioje kalboje, QT IDE), „event binding“, „actions“, „reflections“ ir kitu funkcionalumu.

Todėl, remiantis 1-5 punktais, aplikacijų kūrimas išmaniesiems telefonams taps:

- greitas,
- pigus,
- nebebus susietas nei su konkrečia platforma, nei su konkrečia programavimo kalba.
- Pačių aplikacijų pasiekiamumas taps pasaulinis, todėl turint praktiškai bet kokią išmaniųjų įrenginį bus įmanoma naudotis tokia aplikacija. Be to - pačią aplikaciją sukurti galės bene bet kuris, net ir nedidelius programinius įgūdžius programavime apskritai (pvz. „Python“ abstrakčioje programavimo kalboje), turintis programuotojas.

Todėl esami ir būsimi mobiliųjų OS programuotojai turėtų tikėtis:

- Su konkrečia platforma nesusieto programinio karkaso (angl. a cross-platform framework) - tai jau iš dalies realizuoja QML, ir kompanijai Nokia teliko įgyvendinti savo planus, leisti naudoti QML GUI kūrėjo karkasą ne tik QT C++ aplinkoje, bet parengti QML versijas skirtas

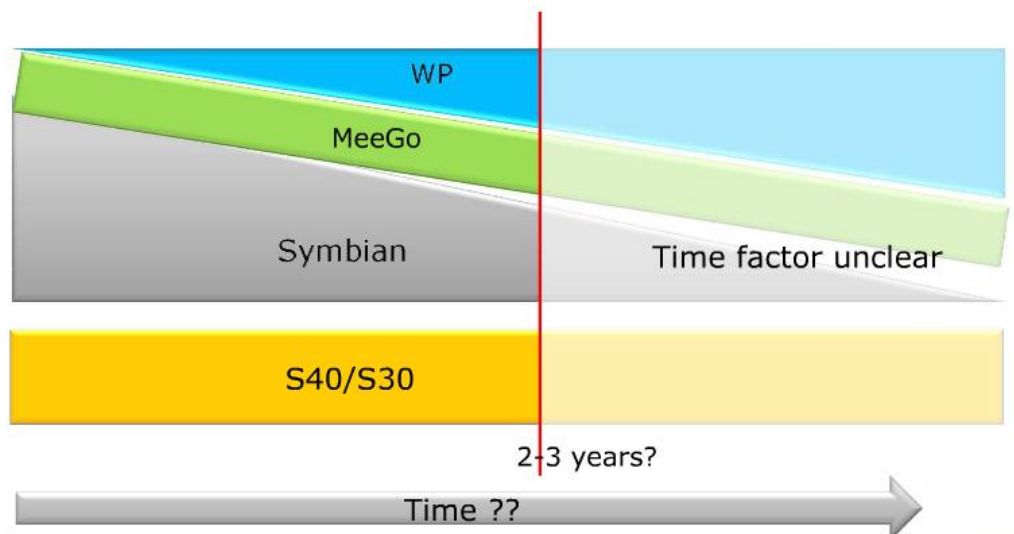
Android OS (t.y. Java kalbos karkasas), iOS (t.y. objektinės C kalbos karkasas) ir Windows Phone (t.y. C# kalbos karkasas)

- Vietoje įprasto programinio kodo rašymo, dažniau naudoti dinaminių kalbų realizacijas, įvairiose platformose naudojamas beveik vienodai – šiuo atveju turime omenyje Python programavimo kalbą ar panašios paskirties programavimo kalbas ir jų karkasus – CPython, Jython, IronPython ir kt.

5.5. Ko tikėtis Symbian ir Series 40 platformų PĮ kūrėjams?

Symbian^3 sulaukė rimtų atnaujinimų 2011 metais – Nokia suplanavo ir 2011 m. vasarą-rudenį pristatė šiuo atnaujinimus ([Hol11]), bei galiausiai pakeitė operacinės sistemos pavadinimą į Nokia Belle, ir didesnių atnaujinimų šiai OS artimiausiais metais, neplanuoja:

- naująją vartotojo sąsają („UI“)
- naujos išvaizdos ir pojūčių „Nokia Store“ (buvusi „Ovi Store“) ir „Nokia maps“ (buvusi „Ovi Maps“) sistemas
- naują lanksčią darbalaukio priedų sistemą (angl. flexible widgets)
- naujas piktogramas
- greitesnę naršyklę, palaikančią daugiau Web 3.0 standartų
- naują navigacinę juostą („Navbar“)
- Symbian^3 sistemą, bet kuriuo atveju, 3-6 metų bėgyje pilnai pakeis Windows Phone 7 ar naujesnės Windows Phone OS. Tačiau čia būtina paminėti porą itin svarbių statistinių faktų, apie kuriuos kalbėjo kompanijos „Digia“ vyr. QT treneris (angl. senior technical trainer) Tuukka Ahoniemi (žr. 14 pav.).



14 pav. Symbian, S40 ir WP7 perspektyvos kompanijoje Nokia

Iš tiesų, 2-3 metų prognozė galbūt ir gali būti nuspėjama, tačiau tolimesniame periode mes turime atsižvelgti į tokius faktus:

Stambiuosiuose prekybos centruose (angl. supermarket), šių dienų išmanieji telefonai su didelėmis nuolaidomis pasirodo po 2-3, o kartais net po 4-5 metų, ir būtent ten, už itin mažą kainą, tikėtina, nemažai klientų tokius telefonus ir įsigys.

Kadangi tokie įrenginiai bus nupirkti, bent 3 iš 6 telefonų vėliau tikrai turės kitą savininką, po 1-3 metų.

Bent 50 proc. tikimybė, kad antrasis tokio telefono savininkas, įsigijęs telefoną, norės naudotis jo specifinėmis funkcijomis, o tai reiškia, kad tai jis darys per tokias sistemas kaip „Nokia Store“, „Nokia Maps“.

Negana to, kadangi toks telefonas bus naudojamas, jame greičiausiai bus rasta ir problemų, spragų, susijusių su jų PI, ir tokie telefonai bus taisomi, vadinasi jiems greičiausiai turės būti paruošti ir pataisų paketai.

Statistiniais duomenimis, šiuo metu pasaulyje vis dar yra naudojama daugiau kaip pusė milijardo mobiliųjų telefonų, kurie yra 10 metų ar senesni. Šį skaičių itin įtakoja ir trečiosios pasaulio šalys. Ir iki šiol yra leidžiamos, kuriamos aplikacijos šiems telefonams.

Todėl, nors Nokia planuose Symbian^3 sistema turėtų tapti tik vidutinės ir žemos klasės, o vėliau tik žemos klasės išmaniųjų telefonų operacinė sistema jau 2-3 metų bėgyje, remiantis realiais pavyzdžiais ir faktais, tikėtina Symbian^3 eros pabaiga yra 2020 metai.

Series 40 tendencijos. Nokia parduoda po 1 mln. S40 įrenginių kasdien, ir šiuo metu jau yra pardavusi virš 600 mln. tokių funkcinių telefonų (angl. feature-phones). Java PI kūrėjai bent artimiausius pora metų nebus užmiršti, ir ši platforma taip pat sulauks didelių atnaujinimų. Kaip pagrindinius paminėsime šiuos:

- Interneto naršyklės funkcionalumo išplėtimas – tai leis nesunkiai įdiegti dalį internetinių aplikacijų (angl. web apps) į S40 OS turintį funkcinį telefoną.
- „Proxy“ naršymo galimybių išplėtimas
- „Touch & type“ funkcija.

Iš šiame poskyryje pateiktų faktų apie Symbian^3 perspektyvas, galime padaryti taip pat porą prielaidų: S40 telefonų pardavimai išsivysčiusiose 1 ir 2 pasaulio šalyse šiuo metu turi bene smarkiausią kritimą, ir per 3-5 metus šiose valstybės telefonai su išmaniosiomis OS užims dominuojančiąją padėtį. Atsilikdamas 3-4 metais, šis procesas vyksta ir trečiojo pasaulio šalyse, todėl S40 platformos eros pabaigos galime tikėtis maždaug 2016-2018 metais.

Reziumuojant galima pasakyti, kad:

- Programinės įrangos kūrimo paklausa Symbian^3 (Nokia Belle) platformai dar ilgą laiką liks pastovi,
- S40 platformoje PĮ kūrimo paklausa mažėja jau šiuo metu,
- Pereinamasis S40 programuotojų persiorientavimo laikotarpis į Symbian^3 (Nokia Belle) programuotojus yra bent 2-3 metai,
- Pereinamasis S40/Symbian^3 (Nokia Belle) programuotojų persiorientavimo laikotarpis į Windows Phone 7 programuotojus yra bent 3-5 metai.

Vadinasi, Symbian^3 (Nokia Belle) programuotojai bent 2-4 metus bus paklausūs, o S40 programuotojai – bent 1-2 metus.

5.6. Ko tikėtis Samsung Bada OS PĮ kūrėjams?

Sulig Bada 2.0 pasirodymu, Samsung turėtų turėti pirmąją savo modernią tikrąją OS, tinkančią naujos kartos išmaniesiems telefonams, bei palaikančią tokias bazines OS funkcijas kaip daugiaprograminis režimas (angl. multi-tasking) ir kt.

Taip pat Samsung stengiasi populiarinti savo platformą skelbdama įvairius konkursus su multi-milijoniniu priziniu fondu programuotojams, siekdama pritraukti programuotojų į šią OS. Ir nors net pati Samsung neketina užleisti visos savo leidžiamųjų išmaniųjų telefonų terpės šiai OS, ir kaip pagrindinę yra pasirinkusi kompanijos Google plėtojamą „Android” OS, sprendžiant iš kompanijos Google politikos, ir tendencijų daryti šią OS vis dažniau uždara bei, galbūt, ateityje paversti tai pilnai komerciniu produktu, „Samsung” kuriama „Bada” OS yra jos saugumo garantas tokiu atveju ir rimtas argumentas OS rinkoje. Todėl, būdama bene sparčiausiai besivystanti mobiliųjų telefonų ir apskritai kitos įrangos kompanija pasaulyje, „Samsung” bet kuriuo atveju toliau investuos į šios OS plėtoją.

Nors šiuo metu šioje OS yra sukurta vos 5000 aplikacijų, tačiau visa tai padaryta šiai sistemai esant dar funkcinio telefono (angl. feature phone) OS stadijoje, kuri nepalaikė tokių esminių funkcijų kaip daugiaprogramis režimas (angl. multi-tasking), dėl privatumo nuostatų neleidžianti pasiekti SMS/MMS serviso, bei kartu iš esmės tebuvo tik tobulesnė J2ME platformos versija.

Ši situacija turėtų tapti smarkiai geresnė, išleidus Bada OS 2.0, kas gali įvykti jau kitais metais. Bet kuriuo atveju, Bada OS aplikacijų paklausa ilgą laiką išliks maža, nors tolygus nedidelis augimas šios OS ir su ja parduodamų įrenginių, yra pastebimas ir šiuo metu 1 mln. vienetų per metų ketvirtį lenkiantis net „Windows Phone 7” įrenginių pardavimus.

Todėl rekomenduojame programuotojams neapsiriboti specializacija vien šios OS terpėje, bet kartu kurti programinę įrangą ir susijusioms OS – perspektyviausia ir panašiausia tokia šiuo metu būtų „Google Android” OS.

5.7. Ko tikėtis Windows Phone, Android ir iOS platformų PĮ kūrėjams?

Šios platformos turi plačias perspektyvas, yra ruošiamos tiek Windows Phone 8, tiek Android 4.0, tiek iOS 5 operacinės sistemos, o šių operacinių sistemų savininkai yra 3 stambiausios pasaulio IT kompanijos – „Microsoft”, „Google” ir „Apple”.

„Microsoft” turi sutartį su „Nokia” – šiuo metu didžiausia mobiliųjų telefonų gamintoja pasaulyje, dėl WP7 ir naujesnių WP operacinių sistemų naudojimo kaip pagrindinių OS Nokia telefonuose.

„Google” bendradarbiauja su Samsung – kuri dar šiemet gali pralenkti „Nokia” ir tapti didžiausia mobiliųjų įrenginių gamintoja pasaulyje, bei yra nusprendusi naudoti „Android” OS kaip pagrindinę OS visuose savo aukštos klasės išmaniuosiuose telefonuose.

„Apple” – pati didžiausia IT kompanija pasaulyje, turinti daugybę lygiagrečių gaminių – iPad, MacBook, iTAB, kurie puikiai parduodami ir siejami su MacOS X ir iOS terpe, todėl ši OS taip pat turi puikias perspektyvas.

5.8. Ko tikėtis Blackberry OS PĮ kūrėjams?

Šios OS PĮ kūrėjai, bei pati OS yra susiradusi specifinę nišą, ypač JAV rinkoje – tai verslo klasės išmanieji solidūs telefonai, orientuoti į pasiturinčius, vidutinio ar vyresnio amžiaus klientus. Šioje terpėje žaidimų ir trimatės grafikos kūrimas ir toliau neturėtų tapti prioritetu, bei, tikėtina, nebus ypatingai tobulinamas. Šios platformos akcentas – įvairių el. pašto, jų konferencijų, konferencinių pokalbių, vaizdo telefonijos protokolai ir jiems skirta gamintojo bei 3-ųjų šalių sukurtos aplikacijos.

REZULTATAI IR IŠVADOS

Šio bakalaurinio darbo metu buvo apžvelgtos visos populiarios žinomos išmaniųjų ir funkcinųjų telefonų operacinių sistemų platformos. Išsamiai išanalizuotos šešios operacinės sistemos – Symbian (Nokia Belle), Android, Windows Phone, Blackberry, iOS ir webOS, kurių naujausios versijos atitinka MOS standartus, todėl yra priskiriamos prie išmaniųjų telefonų modernių operacinių sistemų (MOS). Buvo palyginti šių operacinių sistemų kūrėjų pateikiami kūrėjo paketai (SDK ir IDE), skirti kurti taikomosioms programoms šiose platformose. Buvo išnagrinėtos API sąsajos suteikiamos galimybės programuotojui pasiekti sisteminės funkcijas. Taip pat apžvelgti aparatūriniai reikalavimai šioms OS bei baziniai šių OS telefonų modeliai. Pateikti kiekvienos OS sistemos privalumai, trūkumai, specifinės savybės, programinio kodo pavyzdžiai šiose platformose kuriant GUI terpę. Taip pat sukurta Symbian^3 aplikacija, pavadinimu “Blogink bet kur”, kuri buvo pateikta kompanijai Omnitel.

Remiantis karkasų aplinkų našumo palygimo lentelėmis, Android 2.3 OS paremtame telefone (žr. 1 pav.), naudojant tiek originalų SDK, tiek 3-ųjų šalių daugiaplatforminį karkasą, tiek web-karkasą bei informacija apie karkasų aplinkas (žr. 1.8 poskyrį ir 1.8.1 punktą), prieita tokių išvadų apie karkasų aplinkų našumą ir naudojimo lengvumą: esant dideliame duomenų kiekiui, 3-ųjų šalių aplikacijų karkasas (šiuo atveju buvo panaudotas PhoneGap) yra greitesnis ir lengviau suprantamas nei originalusis SDK (šiuo atveju buvo panaudotas Android SDK), tačiau lėtesnis už web-karkasą (HTML5 ir Javascript), esant tiek mažam, tiek dideliame duomenų kiekiui. Be to web-karkasą suprasti ir naudoti yra lengviausia. Remiantis šaltiniais [Mar11, Yas12], HTML5 turėtų dominuoti jau po artimiausių poros metų lyginant su originaliu platformos SDK populiarumu. Šiame darbe padaryta tokia pat išvada.

Įvertinus OS perprantamumą, veikimo greitumą, našumą, išbaigtumą, OS API patogumą, galima pateikti tokius apibendrinimus:

- Greičiausias PĮ kūrimas su vizualiniais efektais yra su QML, šiuo metu Symbian^3.
- QML minusas – neišbaigtumas: trūksta paties QML karkasų (angl. frameworks).
- QT C++, nors ir turi „Signals & Slots“ funkcinį išplėtimą lyginant su ISO C++, bet vis tiek nėra greita kūrimo aplinka, dėl pačios C++, kaip programavimo kalbos, senumo.
- Programavimo kalbos atžvilgiu – greičiausiai kūrimas vyksta, kuriant Windows Phone 7 aplikacijas, naudojant „Visual Studio 2010“ IDE ir rašant programinį kodą C# programavimo kalba .NET 4.0 karkasui, taip pat Silverlight ir XNA – leidžia itin lengvai kurti žaidimus.
- Perspektyviausiai atrodanti operacinė sistema šiuo metu – Windows Phone 7, jos rinkos dalis turėtų augti geometriškai per ateinančius 3-6 metus.

- Tinkamiausia OS žaidimų kūrimui – Windows Phone 7, dėl šioje OS palaikomo Direct3D grafinio variklio ir jo galimybių. Su Windows Phone 8 išleidimu ir Apollo grafine terpe, ši OS įgaus daugybę iki šiol trūkstamų funkcijų bei atnaujintą grafikos variklį, leidžiantį kurti žaidimus, grafinėmis ir fizikos efektų galimybėmis nenusileidžiančius, esantiems 5 metų senumo stacionariuose kompiuteriuose. Ši terpė itin plačiai išplėtota stacionarių kompiuterių rinkoje ir dėl aplikacijų portabilumo .NET 4.0 platformoje, žaidimų kūrimas šiai terpei turi geriausias galimybes. Antroje vietoje pagal žaidimų kūrimo galimybes yra Android OS, dėl didelių reikalavimų įrenginio centriniui procesoriui, būtino aparatinio būdu spartinamo grafinio procesoriaus su rekomenduojamu OpenGL ES 2.0 palaikymu.

Populiariausia išmaniųjų telefonų operacinė sistema greičiausiai taps ir bent pora metų dominuos – Google Android. Jos pozicijos stiprėja kas mėnesį ir vien dėl tobulesnės programavimo kalbos, tokiai OS kaip Nokia Belle nukonkuruoti Android yra praktiškai neįmanoma. Tačiau dėl savo specifiškumo, Nokia Belle yra kur kas greitesnė ir mažiau resursams reikli operacinė sistema nei Android. Windows Phone 7 taip pat siekia tapti nereiklia resursų atžvilgiu OS. Apple iOS ir RIM Blackberry – tai labiau specializuotos platformos, paremtos labiau šių kompanijų gerbėjais, fanais, kurie išlieka ištikimi šių kompanijų klientai, nors iš techninės pusės šių kompanijų gaminiai nėra tokie tobuli. Tačiau šios kompanijos išsikovojo rinkos dalis, dėl savo specifiškumo ir dėl to, kad buvo kai kurių idėjų pradininkai – pvz. Apple buvo aplikacijų parduotuvės internete pradininkas sukurdamas „AppStore” terpę, kur dėl palankaus finansinio modelio programuotojai buvo labai linkę kurti aplikacijas šiai platformai.

SĄVOKŲ APIBRĖŽIMAI

(mobili) **Operacinė sistema** (OS) – speciali programinė įranga, užtikrinanti vartotojo sąsają ir mobiliojo telefono techninės įrangos, taikomųjų programų bei duomenų valdymą.

(mobili) **Moderni operacinė sistema** (MOS) – OS, su papildomu funkcionalumu, pasižymi „native” pobūdžio kelių užduočių atlikimo sykiu (angl. multi-tasking), foniniais servais (angl. background services), turinti trimatės grafikos aparatūrinio apdorojimo valdymo ir naudojimo bibliotekas, yra lengvai plečiamos.

(mobili) **Funkcinė operacinė sistema** (FOS) – tai nepilnavertė OS, kurioje kelių aplikacijų naudojimas sykiu yra simuliuojamas, nėra pilnaverčio diegimo ir atnaujinimų procesų

Išmanusis telefonas (angl. smartphone) – tai mobilusis telefonas su operacine sistema.

SANTRUMPOS

- ADT – Android kūrėjo įrankiai (angl. Android Development Tools) – Eclipse IDE priedas
- adb – Android derinimo tiltas (angl. Android Debug Bridge)
- API – aplikacijų programinė sąsaja (angl. Application Programmable Interface)
- BREW – bevielių įrenginių dvejetainio kodo paleisties aplinka (angl. Binary Runtime Environment for Wireless)
- DDMS – Dalvik derinimo stebėjimo serveris (angl. Dalvik Debug Monitor Server)
- DEX – Dalvik vykdomasis kodas (angl. Dalvik EXecutable)
- DVM – Dalvik virtuali mašina, naudojamas kompanijos „Google“ (angl. Dalvik Virtual Machine)
- FOS – funkcinė operacinė sistema (angl. Feature Operating System)
- GUI – grafinė vartotojo sąsaja (angl. Graphic User Interface)
- IDE – integruota kūrimo aplinka (pvz. NetBeans, Eclipse ir kt.) (angl. Integrated Developer Environment)
- J2ME – Java Platform, Micro Edition
- J2SE – Java Platform, Standart Edition
- J2EE – Java Platform, Enterprise Edition
- JDK – Java kūrėjo paketas (angl. Java Developer Kit)
- JRE – Java paleisties aplinka (angl. Java Runtime Environment)
- JVM – Java virtuali mašina, naudojamas kompanijos „Oracle“ (angl. Java Virtual Machine)
- MOS – moderni operacinė sistema (angl. Modern Operating System)
- MVC – modelių-vaizdų valdiklis (angl. Model View controler)
- OS – operacinė sistema (angl. Operating System)
- OTASL – PĮ atnaujinimai nuotoliniu būdu (angl. Over the air software loading)
- PDA - asmeninis skaitmeninis pagalbininkas (angl. Personal digital assistant)
- PĮ – programinė įranga (angl. Software)
- SDK – PĮ kūrėjo paketas(angl. Software Developer Kit)
- UI – vartotojo sąsaja (angl. Graphic User Interface)
- VM – Virtuali mašina (angl. Virtual Machine)
- WP7 – Windows Phone 7 (operacinės sistemos pavadinimas)

ŠALTINIAI

Knygos, konferencijų medžiaga, dokumentacijos:

- [Che10] Lu Cheng. Analysis and Comparison with Android and iPhone Operating System. 2010 kovas.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.eecs.ucf.edu/~dcm/Teaching/COP5611Spring2010/Project/AmberChang-Project.pdf>>
- [ECM+12] Jordán Pascual Espada, Rubén González Crespo, Oscar Sanjuán Martínez, B. Cristina Pelayo G-Bustelo, Juan Manuel Cueva Lovelle. Extensible architecture for context-aware mobile web applications. 2012 vasaris.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.sciencedirect.com/science/article/pii/S0957417412004149>>
- [Goo10] Google Inc. Android 2.3 Compatibility Definition. Google, 2010.
[žiūrėta 2011-05]. Prieiga per internetą:
<<http://source.android.com/compatibility/2.3/android-2.3.3-cdd.pdf>>
- [Goo10b] Google Inc. Android Developers. DDMS, 2010.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://developer.android.com/guide/developing/debugging/ddms.html>>
- [Goo10c] Google Inc. Android Developers. ADT Plugin, 2010.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://developer.android.com/sdk/eclipse-adt.html>>
- [HBH12] Daniel Hilkert, Alexander Benlian, Thomas Hess. The Openness of Smartphone Software Platforms – A Framework and Preliminary Empirical Findings from the Developers' Perspective. 2012
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.user.tu-berlin.de/komm/CD/paper/090322.pdf>>

- [Yas12] Patel Yashodhar. Comparing performance of applications written in smartphone development environments. 2012 sausis.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://sdsu-dspace.calstate.edu/xmlui/handle/10211.10/1694>>
- [LY09] Feida Lin, Weiguo Ye. Operating System Battle in the Ecosystem of Smartphone Industry. Konferencijos data: 2009 gegužės 16-17, p. 617-621
[žiūrėta 2012-05]. Prieiga per internetą:
<http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5175193>
- [OT10] Julian Ohrt, Volker Turau. Cross-Platform Development Tools for Smartphone Applications. 2010 lapkritis.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06178193>>
- [SGE09] A. Savoldi, P. Gubian, I. Echizen. A Comparison between Windows Mobile and Symbian S60 Embedded Forensics.
Konferencijos data: 2009 rugsėjo 12-14, p 546-550
[žiūrėta 2012-05]. Prieiga per internetą:
<http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5337370>
- [Sha11] Kavita Sharma. Android in opposition to iPhone. Online Journal, Vol. 3 No. 5 May 2011.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.enggjournals.com/ijcse/doc/IJCSE11-03-05-102.pdf>>
- [TSL12] David Tilson, Carsten Sørensen, Kalle Lyytinen. Change and Control Paradoxes in Mobile Infrastructure Innovation. 2012.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://digitalinfrastructures.org/publications/TilsonSorensenLyytinen2012.pdf>>

Elektroniniai teikiniai:

- [And11] AndroidLib. Android Market statistics
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.androlib.com/appstats.aspx>>
- [App11] Apple. Apple's App Store Downloads Top 10 Billion
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.apple.com/pr/library/2011/01/22appstore.html>>
- [App12] Apple, Inc. XCode
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://developer.apple.com/xcode/>>
- [Bai10] Edward C. Baig. GetJar offers non-smartphone owners app ability. USA today. 2 October, 2010.
[žiūrėta 2012-05]. Prieiga per internetą:
< http://www.usatoday.com/tech/columnist/edwardbaig/2010-02-11-Baig11_ST_N.htm >
- [Big11] John Biggs. Samsung App Store Hits 100 Million Download Mark. Mobile crunch. March 24th, 2011.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.mobilecrunch.com/2011/03/24/samsung-app-store-hits-100-million-download-mark/>>
- [Cam12] Wilian Campinas. Number of smartphones in the world is expected to reach 1 billion by 2016, consultancy projects. Broadband mobile opportunities. February 13, 2012
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://broadband-mobile-oportunities.blogspot.com/2012/02/number-of-smartphones-in-world-is.html>>
- [CE05] Phanfare CEO, Andrew Erlichson. GUI Development on the Mac vs. the PC. Phanfare blog. October 14th, 2005.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://blog.phanfare.com/2005/10/gui-development-on-the-mac-vs-the-pc/>>

- [Dkd10] DKgADGET. Google says hardware requirement for Android 2.3 are similar to Android 2.2 but manufacturers are confused! December 9, 2010.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://dkgadget.com/google-hardware-requirement-android-2-3-similar-android-2-2-manufacturers-confused/>>
- [DM10] An Introduction to webOS. Slideshares. Jun 30, 2010.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.slideshare.net/kpdecker/an-introduction-to-webos>>
- [Fir11] Max Firtman. Symbian Anna Browser & HTML5: is it the evolution expected? Mobilexweb. April 14th, 2011.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.mobilexweb.com/blog/symbian-anna-browser-html5>>
- [Gar10] Gartner. Gartner Says Worldwide Mobile Device Sales to End Users Reached 1.6 Billion Units in 2010; Smartphone Sales Grew 72 Percent in 2010
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.gartner.com/it/page.jsp?id=1543014>>
- [Goo11c] Google, Inc. Android Open Source Project.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://source.android.com/compatibility/index.html>>
- [Goo11d] Google, Inc. Android Open Source Project. Compatibility Test Suite : How does the CTS work?
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://source.android.com/compatibility/cts-intro.html>>
- [Gos08] Gossamer threads. Python for BlackBerry. March, 2008.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.gossamer-threads.com/lists/python/python/629959>>

- [Gsm10] Gsmarena. Apple iPhone 4.
[žiūrēta 2012-05]. Prieiga per internetą:
<http://www.gsmarena.com/apple_iphone_4-3275.php>
- [Ham10] Carol Hamer. Blackberry game development
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.coderanch.com/t/511075/BlackBerry/Incubator/Game-development>>
- [Hin10] Jason Hiner. Android 2.3 'Gingerbread': Feature list, screenshots, and video.
Techrepublic. December 7, 2010
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.techrepublic.com/blog/hiner/android-23-gingerbread-feature-list-screenshots-and-video/7099>>
- [Hol11] Thom Holwerda. Nokia Talks About Future of Symbian, Qt, Series 40. OS news.
28th Mar 2011.
[žiūrēta 2012-05]. Prieiga per internetą:
<http://www.osnews.com/story/24577/Nokia_Talks_About_Future_of_Symbian_Qt_Series_40>
- [Ind10] The Independent. GetJar smartphone app shop closes in on a billion downloads. 11 January 2010.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.independent.co.uk/news/media/getjar-smartphone-app-shop-closes-in-on-a-billion-downloads-1864736.html>>
- [Yin11] Wesley Yin-Poole. Total Apple iOS sales: 189 million. Eurogamer. 21 April, 2011.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.eurogamer.net/articles/2011-04-21-total-apple-ios-sales-189-million>>
- [Ken10] Project Kenai. Android plugin for netbeans. 2010
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://kenai.com/projects/nbandroid>>

- [Kyl09] Kyle. First leaked screenshots of Samsung Bada OS. Gsmarena blog. December 14th, 2009.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://blog.gsmarena.com/first-leaked-screenshots-of-samsung-bada-os/>>
- [Mac12a] Macroaxis. Nokia Number of Employees.
[žiūrēta 2012-05]. Prieiga per internetą:
<http://www.macroaxis.com/invest/ratio/NOK--Number_of_Employees>
- [Mac12b] Macroaxis. Microsoft quote and profile.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.macroaxis.com/invest/market/MSFT>>
- [Mar11] Matt Marshall. How HTML5 Will Kill the Native App, 2011.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://venturebeat.com/2011/04/07/how-html5-will-kill-the-native-app/>>
- [Mas11] Mashable. Microsoft Shares WP7 Stats, Takes a Few Jabs at Other Platforms.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://mashable.com/2011/03/31/microsoft-wp7-stats/>>
- [McI10] Kyle McInnes. BlackBerry 6 Review and Guide: FAQ, Tips and Tricks and Impressions. Blackberry cool. August 13, 2010.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.blackberrycool.com/2010/08/13/blackberry-6-review-and-guide-faq-tips-and-tricks-and-impressions/>>
- [Mic10] Jason Mick. UPDATED: Benchmark: iPhone 4 HTML5 Performance is Far From "Magical". Dailytech. September 21, 2010.
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.dailytech.com/UPDATED+Benchmark+iPhone+4+HTML5+Performance+is+Far+From+Magical/article19681.htm>>

- [Mil09] Paul Miller. What Apple could learn from Palm's webOS. Engadget. January 15th 2009.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.engadget.com/2009/01/15/what-apple-could-learn-from-palms-webos/2>>
- [Mob12a] MobileBurn. What is "S40"?
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.mobileburn.com/definition.jsp?term=S40>>
- [Mob12b] Mobi Thinking. Global mobile statistics 2012: all quality mobile marketing research, mobile Web stats, subscribers, ad revenue, usage, trends... February, 2012.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats>>
- [Naz09] Nazmul. Creating your first GUI BlackBerry App – Tutorial. Developer life. September 24th, 2009.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://developerlife.com/tutorials/?p=738>>
- [Nok11] Nokia. Qt Training and MoMo in Tallinn. Nokia blogs. April 18, 2011.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://blogs.nokia.com/nordicblog/news/qt-training-and-momo-in-tallinn/>>
- [Omn12] Omnitel. 2012 m. pirmąjį ketvirtį „Omnitel“ mobiliųjų duomenų verslas augo 13 proc. Balandžio 19 d., 2012 m.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.omnitel.lt/ziniasklaidai/2012-m-pirmaji-ketvirti-omnitel-mobiliuju-duomenu-verslas-augo-13-proc-/56773>>
- [Poc10] [žiūrėta 2012-05]. Prieiga per internetą:
<http://www2.pocketgamer.co.uk/r/PG.Biz/Nokia+World+2010/news.asp?b=related&c=23490>

- [Rea09] Marguerite Reardon. GetJar: The unknown app store leader. Cnet News. November 6, 2009.
[žiūrėta 2012-05]. Prieiga per internetą:
<http://news.cnet.com/8301-30686_3-10392095-266.html>
- [Spe12] Alexander Spektor. Apple Becomes World's Largest Smartphone Vendor in Q4 2011. Strategy analytics. January 26, 2012.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.strategyanalytics.com/default.aspx?mod=pressreleaseviewer&a0=5170>>
- [Spr08] Sprint. BlackBerry SP8330RIM.
[žiūrėta 2012-05]. Prieiga per internetą:
<http://developer.sprint.com/view_devices.do?device=79541>
- [Sav11] Vlad Savov. Google Ice Cream Sandwich coming in Q4 2011 to smartphones and tablets alike. Engadget. May 10th 2011.
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.engadget.com/2011/05/10/google-announces-ice-cream-sadwich-for-q4-2011-for-smartphones/>>
- [Sta12] Vaidotas Stanevičius. Windows Phone marketplace plečiasi „greičiau” už Android, bet lėčiau už iOS. FWD.LT. 2012 birželio 6.
[žiūrėta 2012-06-06]. Prieiga per internetą:
<<http://www.fwd.lt/2012/naujienos/windows-phone-marketplace-pleciasi-greiciau-uz-android-bet-leciau-uz-ios>>
- [Wes08] Tony Westbrook. Creating a blackberry game – part 1. Tony Westbrook dot com. MAY 03, 2008
[žiūrėta 2012-05]. Prieiga per internetą:
<<http://www.toniwestbrook.com/archives/69>>

- [Wil11] Adam Wills. Blackberry Development – Increase Your Capability and Reputation. Docstoc. February 17, 2011
[žiūrēta 2012-05]. Prieiga per internetą:
<<http://www.docstoc.com/docs/71747474/Blackberry-Development-%E2%80%93-Increase-Your-Capability-and-Reputation>>
- [Wik12] Wikipedia. List of mobile software distribution platforms.
[žiūrēta 2012-05]. Prieiga per internetą:
<http://en.wikipedia.org/wiki/List_of_digital_distribution_platforms_for_mobile_devices>

```

<!DOCTYPE HTML>
<html>
<head>
<meta name="viewport" content="width=320; user-scalable=no" />
<meta http-equiv="Content-type" content="text/html; charset=utf-8">
<title>PhoneGap</title>
<link rel="stylesheet" href="master.css" type="text/css" media="screen"
      title="no title" charset="utf-8">

<script type="text/javascript" src="jquery.min.js"></script>
<script type="text/javascript" charset="utf-8" src="phonegap-
0.9.3.js"></script>
<style type="text/css">
.View {
    background-color: blue;
    width: 320px;
    height: 500px;
}

.StartLine {
    margin-top: 19px;
    background-color: white;
    height: 1px;
}

.Rect {
    background-color: white;
    position: absolute;
    margin-top: 1px;
    margin-left: 20px;
    width: 60px;
    height: 60px;
}

.clearFix {
    clear: both;
    height: 0;
    width: 100%;
    display: block;
}
</style>
<script><!--
var elem;
var context;
var x = 0;
var y = 450;
var dx = 1;
var WIDTH = 320;
var HEIGHT = 450;
var intervalId;
var count = 0;
var start;
$(document).ready(function() {

    elem = document.getElementById('myCanvas');
    if (!elem || !elem.getContext) { return; }
    // Get the canvas 2d context.

    context = elem.getContext('2d');
    if (!context) { return; }

});
function startAnimation(){
    count = 0;
    var elem = document.getElementById('myCanvas');
    if (!elem || !elem.getContext) { return; }

```

```

        // Get the canvas 2d context.
        var context = elem.getContext('2d');
        if (!context) { return; }
        var dir = 0;
        var d = new Date();
        var curr_sec = d.getSeconds();
        var curr_msec = d.getMilliseconds();

        start = (curr_sec * 1000) + curr_msec;
        start = new Date().valueOf();
        intervalId = setInterval(drawRect,0);

    }
    function ShowTimeInterval()
    {
        var e = new Date();
        var end = new Date().valueOf();
        var result = "Start Timestamp: "+start+" <br /> Interval: "+( end
- start) + "<br />";
        $('#Result').html(result);
    }
    function drawRect(){
        context.clearRect(0,0,320,450);
        context.fillStyle = '#00f'; // blue
        if (x + dx > WIDTH || x + dx < 0){
            dx = -dx;
            count++;
            if(count > 5){
                clearInterval(intervalId);
                ShowTimeInterval();
            }
        }
        x += dx;
        context.fillRect(20, x, 40, 40);
    }--></script>
</head>
<body id="stage" class="theme">
<h1>Welcome to PhoneGap!</h1>
<h2>this file is located at assets/index.html</h2>
<div id="Result"> Time Interval</div>
<a href="#" onclick="startAnimation();">Click HERE to Start</a>
<div class="clearFix"></div>
<canvas id="myCanvas" width="320" height="450">
This Browser does not support Canvas.
</canvas>
</body>
</html>

```

2.1. Symbian 10.1 (Nokia Belle)

Kuriant Nokia Belle PĮ per QT, nenaudojant QML, yra Drag n' drop įrankis.

Symbian^3 GUI sąsajos kūrimas QML kalboje naudojant QT Quick.

```
import QtQuick 1.0
Rectangle {
width: 200; height: 200
color: "lightblue"
Text {
id: helloText
anchors.horizontalCenter: parent.horizontalCenter
font.pixelSize: parent.height / 10
font.bold: true
text: "Meet QML!"
}
Image {
id: helloImage
anchors.centerIn: parent
source: "icons/qt_logo.png"
}
MouseArea {
anchors.fill: parent
onClicked: {
helloImage.visible = false;
helloText.text = "Bye-bye picture!";
}
}
}
```

Rezultatas būtų toks (žr. 15 pav.):



15 pav. QT rezultatas

2.2. Windows Phone 7

Kuriant WP7 per Visual Studio 2010 yra – yra Drag n' drop įrankis. Pats kodas deklaratyvus YAML pagrindu. Po juo „events“ ir „actions“ aprašyti C# kalboje.

2.3. Google Android 2.3

Google App Inventor IDE aplinka turi funkcionalų Drag n' drop kūrimo įrankį. Galima rašyti GUI kodą ir pačiam programuotojui, tačiau kodas nėra trumpas ir nepasižymi galimybėmis aprašyti GUI jo minimaliu kiekiu, todėl jo nepateiksime.

2.4. Apple iOS 5.1

Xcode 4 naudoja funkcionalų drag n' drop pobūdžio GUI kūrimo įrankį. Galima rašyti GUI kodą ir pačiam programuotojui, tačiau kodas nėra trumpas ir nepasižymi galimybėmis aprašyti GUI jo minimaliu kiekiu, todėl jo nepateiksime.

2.5. HP webOS 2

GUI sąsajos kūrimas vyksta naudojant Mojo šabloną ir jo galimybes. Šis šablonas turi drag n' drop funkcijas, taip pat kodas rašomas ir HTML + Javascript būdu.

Pateiksime pavyzdinį GUI sąsajos kūrimą, naudojant Javascript ir HTML.

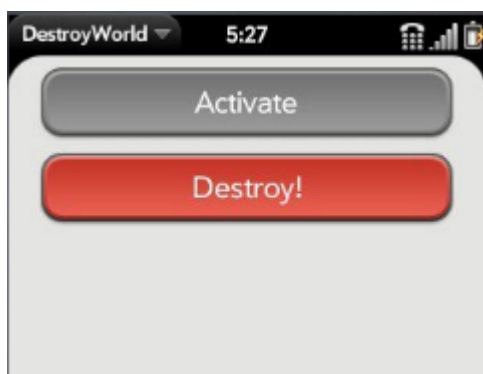
1. HTML kodo dalis:

```
<div id="activate-button" class="palm-button">Activate</div>  
<div id="destroy-button" class="palm-button negative">Destroy!</div>
```

2. Javascript kodo dalis:

```
ActivateAssistant.prototype.setup = function() {  
  this.controller.listen(  
    "destroy-button", Mojo.Event.tap, this.destroy.bind(this));  
};  
  
ActivateAssistant.prototype.destroy = function(event) {  
  console.log("destroying with event " + event);  
};
```

3. Gautas rezultatas vizualiai atrodo taip (žr. 16 pav.):



16 pav. Pavyzdinės webOS GUI aplikacijos rezultatas

2.6. RIM Blackberry 6.0

Blackberry GUI kūrimas vyksta MIDP specifikacijų arba RIM UI Api pagrindu.

Žemiau pateikiamas GUI kūrimo pavyzdys naudojant RIM UI Api.

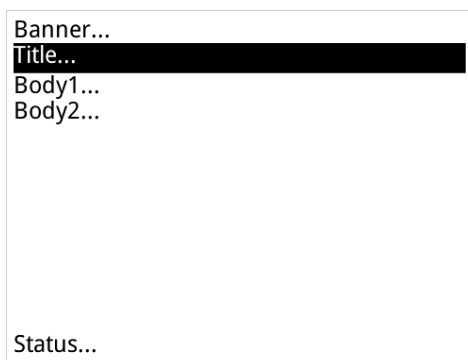
Aplikacijos inicializavimas su GUI sąsaja:

```
publicclass HelloWorldApp extends UiApplication {  
    // main method  
    publicstaticvoid main(String[] args) {  
        HelloWorldApp theApp = new HelloWorldApp();  
        UiApplication.getUiApplication().pushScreen(new MyScreen());  
        theApp.enterEventDispatcher();  
    }  
    ..  
}
```

Pavyzdinis kodas, naudojantis GUI sąsajos funkcionalumą:

```
publicclass MyScreen extends MainScreen {  
    public MyScreen() {  
        // create a label  
        LabelField title = new LabelField("Title...",  
            LabelField.ELLIPSIS | LabelField.USE_ALL_WIDTH);  
  
        // create a label  
        LabelField banner = new LabelField("Banner...",  
            LabelField.ELLIPSIS | LabelField.USE_ALL_WIDTH);  
<..> // Status..  
  
        // create a label  
        LabelField body1 = new LabelField("Body1...",  
            LabelField.ELLIPSIS | LabelField.USE_ALL_WIDTH);  
<..> // Body2..  
  
        VerticalFieldManager vfm = new VerticalFieldManager(Field.FIELD_VCENTER);  
        vfm.add(body1);  
        vfm.add(body2);  
        add(vfm);  
        setTitle(title);  
        setBanner(banner);  
        setStatus(status);  
    }  
}
```

Vizualiai atrodo štai taip (nuotrauka iš „Blackberry device simulator“, žr. 17 pav.):



17 pav. Blackberry GUI sąsajos vaizdo pavyzdys