

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Dominykas Barisas

**Mobiliųjų įrenginių dinaminės vartotojo sąsajos
kūrimas ir tyrimas**

Magistro darbas

Darbo vadovas

prof. E. Bareiša

Kaunas, 2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Dominykas Barisas

**Mobiliųjų įrenginių dinaminės vartotojo sąsajos
kūrimas ir tyrimas**

Magistro darbas

Vadovas

prof. E. Bareiša
2007-05

Recenzentas

2007-05

doc. A. Riškus

Atliko

IFM-1/2 gr. stud.
Dominykas Barisas
2007-05-20

Kaunas, 2007

SUMMARY

Evolving software development for mobile devices brings an increasing importance to develop reliable, user-friendly applications and do it in as short amount of time as possible.

Very important is graphical user interface design, it's functionality, simplicity and fault free operation. Dynamic and fast low-level user interface development allows to significantly reduce the amount of time that development takes, provides fast content updates requiring less user endeavour, gives easy access to the most of the mobile device provided functionality. Furthermore, testing becomes easier and lasts shorter.

Dynamic low-level graphical user interface development and testing are analyzed in the project, established goals and proposed solutions for their implementation. Dynamic user interface was defined using XML, advantages and disadvantages of such architecture were listed and conclusions provided.

SANTRAUKA

Tobulėjant informacinėms technologijoms vis didesnis programinės įrangos procentas kuriama mobiliųjų technologijų vartotojams. Populiarėjant mobiliųjų įrenginių programinei įrangai, darosi vis svarbiau sukurti patikimas, gerai vartotojui suprantamas programas ir tai padaryti kiek įmanoma greičiau.

Gan svarbus yra vartotojo sąsajos kūrimas, jos funkcionalumas, paprastumas ir teisingas veikimas. Dinaminis vartotojo sąsajos kūrimas leidžia žymiai sumažinti programinei įrangai sukurti reikalingą laiką, suteikia mažai pastangų reikalaujantį vartotojo sąsajos atnaujinimą, greitą priėjimą prie daugumos mobilaus įrenginio teikiamų funkcijų. Taip pat palengvėja testavimo proceso automatizavimas, leidžia sumažinti bandymų trukmę

Šiame darbe nagrinėjamas mobiliųjų įrenginių dinaminis grafinės vartotojo sąsajos kūrimas ir jos testavimas, nustatomi tikslai ir galimas jų įgyvendinimas. Darbe dinaminės vartotojo sąsajos aprašymui buvo pasirinkta XML kalba, įvardinti tokios architektūros privalumai ir problemos ir pateiktos išvados.

TURINYS

1	ĮVADAS	1
2	ANALITINĖ DALIS	2
2.1	Tikslai	2
2.2	Problemos	2
2.3	Egzistuojantys sprendimai	4
2.3.1	XUL	4
2.3.2	BambooKit GUI.....	5
2.3.3	Thinlet.....	8
2.4	J2ME pasirinkimo pagrindimas	8
2.5	J2ME dinaminė vartotojo sąsaja	9
2.5.1	Vartotojo sąsajos elementai	9
2.5.2	Standartinių komponentų ir Canvas grafinių elementų palyginimas.....	10
2.5.3	Canvas naudojimo privalumai	12
2.5.4	Canvas ir XML	12
2.5.5	XML skaitymas ir skaitymo greitis	13
2.6	Naudojamos technologijos.....	14
2.6.1	Record Management System	14
2.6.2	SyncML protokolas.....	15
2.7	J2ME testavimo galimybių analizė.....	16
2.7.1	Grafinės vartotojo sąsajos testavimas	16
2.7.2	Testinių atvejų generavimas	17
2.7.3	Testinių atvejų vykdymas	18
2.7.4	Mobiliųjų įrenginių programų testavimo įrankiai.....	19
2.8	Analizuojamų technologijų pritaikymas	21
3	PROJEKTINĖ DALIS	22
3.1	Sistemos aprašymas	22
3.1.1	Sukurta sistema	22
3.1.2	Funkcionalumas	22
3.1.3	Adresatas.....	23
3.2	Sistemos realizacija.....	23

3.2.1	Panaudojimo atvejai.....	24
3.3	Architektūra	26
3.3.1	Apžvalga	26
3.3.2	Sistemos statinis vaizdas.....	27
3.4	Programų sistemos diegimas bei administravimas	28
3.4.1	Reikalinga techninė ir programinė įranga.....	28
3.4.2	Pagrindinės problemos.....	29
3.4.3	Sistemos administravimo Web sąsaja.....	30
3.5	Sistemos patobulinimo apžvalga	31
4	TYRIMO DALIS	32
4.1	Tikslai	32
4.2	Problemos	32
4.3	Sprendimo būdų apžvalga.....	33
4.4	Sistemos struktūra.....	33
4.4.1	Bendras vaizdas	34
4.4.2	J2ME taikomosios programos architektūriniai sprendimai	34
4.5	Dinaminės vartotojo sąsajos elementų aprašymas.....	35
4.5.1	Abstraktus grafinis elementas, elementų funkcionalumas.....	36
4.5.2	Langas	38
4.5.3	Mygtukas	39
4.5.4	Panelis	39
4.5.5	Paveikslukas	40
4.5.6	Įvedimo laukelis.....	41
4.5.7	Tekstas	42
4.5.8	Judantis tekstas	43
4.5.9	Progreso juosta.....	44
4.5.10	Parinkčių mygtukai	45
4.5.11	Funkcinis mygtukas	46
4.6	Vartotojo sąsajos funkcionalumo aprašymas.....	46
4.7	Programos veiksmų seka	48
4.8	Sistemos privalumai.....	49
4.8.1	Galimybė kurti nestandartinius elementus.....	49
4.9	Testavimo galimybių apžvalga	50
4.9.1	J2ME vienetų testavimas	50
4.9.2	Vartotojo sąsajos testavimas.....	51

5	EKSPERIMENTINĖ DALIS	53
5.1	XML analizatoriaus veikimo greičio tyrimas ir galimi patobulinimai	53
5.2	Dinaminės grafinės vartotojo sąsajos elementų piešimo tyrimas	54
5.3	Testavimo galimybių tyrimas	55
6	ATEITIES PATOBULINIMAI	58
6.1	Sinchronizacija su serveriu	58
6.2	Kiti patobulinimai	58
7	IŠVADOS	60
8	LITERATŪRA	61
9	TERMINŲ IR SANTRUMPŲ ŽODYNAS	64
10	PRIEDAI.....	65
	A XUL vartotojo sąsajos aprašai	65
	B SyncML protokolo pranešimo pavyzdys	66
	C Vartotojo sąsajos aprašas XML kalba	67
	D Konferencijų medžiaga.....	70

1 ĮVADAS

Šis darbas yra dinaminės grafinės vartotojo sąsajos kūrimo mobiliuosiuose įrenginiuose tyrimas, įvardinami privalumai ir kylančios problemos, apžvelgiamos tokios architektūros programų testavimo galimybės, pateikiami tyrimo rezultatai ir išvados.

Tobulėjant informacinėms technologijoms vis didesnis programinės įrangos procentas kuriama interneto ir mobiliųjų technologijų vartotojams. Populiarėjant mobiliųjų įrenginių programinei įrangai, darosi vis svarbiau sukurti patikimas, gerai vartotojui suprantamas programas ir tai padaryti kiek įmanoma greičiau.

Gan svarbus yra vartotojo sąsajos kūrimas, jos funkcionalumas, paprastumas ir teisingas veikimas. Dinaminis vartotojo sąsajos kūrimas leidžia žymiai sumažinti programinei įrangai sukurti reikalingą laiką, suteikia mažai pastangų reikalaujantį vartotojo sąsajos atnaujinimą, greitą priėjimą prie daugumos mobilaus įrenginio teikiamų funkcijų. Taip pat palengvėja testavimo proceso automatizavimas, leidžia sumažinti bandymų trukmę.

Šiame darbe nagrinėjamas mobiliųjų įrenginių dinaminis grafinės vartotojo sąsajos kūrimas ir jos testavimas, nustatomi tikslai ir galimas jų įgyvendinimas. Darbe dinaminės vartotojo sąsajos aprašymui buvo pasirinkta XML kalba, įvardinti tokios architektūros privalumai ir problemos.

2 ANALITINĖ DALIS

2.1 Tikslai

Vienas pagrindinių tikslų yra dinaminės vartotojo sąsajos kūrimas mobiliosioms platformoms, suteikiant priėjimą prie beveik visų mobilaus įrenginio teikiamų funkcijų. Pagrindiniai dinaminei programinei įrangos vartotojo sąsajai keliami reikalavimai yra šie:

- ✓ Žemo lygio taikomosios programos sukūrimas, kuri skaitytų ir nagrinėtų XML failų turinį ir pagal tai kurtų vartotojo sąsają
- ✓ Lango ir kitų elementų aprašymas XML kalba, perėjimas tarp skirtingų langų, grįžimas į ankstesnįjį
- ✓ Palyginti paprastos ir dinaminės vartotojo sąsajos vėlinimą, t.y. laiko trukmę nuo vartotojo veiksmo pradžios iki atvaizduoto rezultato (pavyzdžiui perėjimas iš vieno lango į kitą)
- ✓ Atlikti XML failų skaitymo ir nagrinėjo greičio testus, siūlyti būdus proceso pagreitinimui
- ✓ Remiantis ta pačia XML paremta programos architektūra, atlikti galimų testavimo būdų analizę

2.2 Problemos

Programavimo technologijos mobiliems įrenginiams nėra orientuotos į konkrečią aparatūrinę architektūrą ir gali būti taikomos įvairiems mobiliems įrenginiams, turintiems J2ME palaikymą. Mobilųjų technologijų standartai neapibrėžia kaip aparatūros gamintojai turėtų realizuoti technologijos palaikymą. Todėl grafinė vartotojo sąsaja skirtingų gamintojų įrenginiuose skiriasi viena nuo kitos. Kuriant vartotojo sąsają reikia atsižvelgti į įrenginių skirtumus, tokius kaip skirtingas ekrano dydis ar skirtingi mygtukai naudojami tam pačiam funkcionalumui pasiekti. Todėl vartotojo sąsaja turi būti išbandyta visuose įrenginiuose, į kuriuos yra orientuota programa.

Grafinės vartotojo sąsajos ir programos operacijų scenarijų testavimas mobiliems įrenginiams dažniausiai yra atliekamas rankiniu būdu. Testavimą galima automatizuoti iš dalies, tačiau visiškas testavimo automatizavimas mobilųjų įrenginių vartotojo sąsajai yra praktiškai neįmanomas.

Rinkoje egzistuoja labai didelė mobiliųjų įrenginių įvairovė, ir kiekvienas įrenginys skiriasi savo išvaizda, funkcinėmis galimybėmis, ekrano dydžiu ir daugeliu kitų savybių [1].

Senesni telefonai turi MIDP 1.0 palaikymą, tačiau čia aptarsime tik naujesnius įrenginius, kurie palaiko MIDP 2.0. Palyginimui su MIDP 1.0, MIDP 2.0 biblioteka turi daugiau klasių ir metodų, užima truputį mažiau atminties vietos, palaiko permatomus paveikslukus.

Kiekvienam mobiliajam įrenginiui ar jų grupei yra būdingas tam tikras savybių rinkinys. Keleto tokių savybių apžvalga pateikiama (1 lentelė).

1 lentelė. Mobiliųjų įrenginių charakteristikų aprašymas

Savybė	Aprašymas
Ekranų dydis	Paprastuose telefonuose dažniausiai pasitaikančios ekrano rezoliucijos yra 128x128, 128x160, 176x220, 240x320 pikseliai. 2 pav. pateikiamas keleto įrenginių pavyzdžiai.
Klavišų funkcionalumas	Skirtingų gamintojų įrenginiai turi papildomus funkcinius klavišus tam pačiam funkcionalumui pasiekti (2 pav.), arba esamų tų pačių klavišų kodai skiriasi [2].
Foto kamera	Daugelis įrenginių turi integruotą kamerą, kurios pagalba galima tiek daryti nuotraukas tiek filmuoti.
Bluetooth technologija	Naudojama bevieliui duomenų perdavimui, integruota į daugumą šiuolaikinių įrenginių.
WI-FI palaikymas	Nemažas skaičius Symbian ir Windows Mobile įrenginių turi integruotą WI-FI palaikymą
GPS imtuvas	Pradeda atsirasti vis daugiau įrenginių turinčių integruotą arba prijungtą GPS imtuvą.



1 pav. Skirtingų ekrano rezoliucijų įrenginių pavyzdžiai

Įrenginio funkcionalumas bei savybės neretai priklauso nuo įrenginio paskirties. Pavyzdžiui, įrenginiai skirti rašyti dokumentams bei el. laiškam turi didesnę ekrano rezoliuciją ir patogesnę klavišų išsidėstymą.



2 pav. Įrenginių su skirtingu klavišų kiekiu bei funkcionalumu pavyzdžiai

2.3 Egzistuojantys sprendimai

2.3.1 XUL

XUL yra Mozilla XML-paremta vartotojo sąsajos kalba [3], kuri leidžia kurti daugelį reikalingų funkcijų turinčias nuo platformos nepriklausančias programas, kurios veikia tiek prisijungus, tiek neprisijungus prie interneto. Šios programos gali būti lengvai keičiamos pridendant papildomą tekstą, grafinius elementus, keičiant išdėstymą.

Trumpai apžvelgsime tokios sąsajos teikiamus privalumus:

✓ **Galinga žymenų kalba**

XUL pagrindinis tikslas yra daugiaplatformių programų kūrimas. Lyginant su DHTML, kuris yra skirtas internetinių puslapių kūrimui, XUL skiriasi tuo, kad yra orientuota į paprastų programų vartotojo sąsajos kūrimą: langai, tekstas, mygtukai ir kt.

✓ **Paremta egzistuojančiais standartais**

XUL yra XML kalba paremta W3C standartu XML 1.0. Programos, parašytos XUL kalboje yra paremtos papildomu W3C standartais HTML 4.0, CSS 1 ir 2, DOM lygiais 1 ir 2, Javascript 1.5.

✓ **Platformos pernešamumas**

Kaip ir HTML, XUL yra suprojektuota būti nuo platformos nepriklausoma, tai leidžia lengvą programų pernešamumą visoms platformoms, kuriose veikia Mozilla. Turint omenyje, kad Mozilla yra palaikoma daugelyje platformų, XUL kalbos naudojimas programų kūrimui gali greitai tapti labai populiarus, nes jis teikia aukšto lygio vartotojo sąsajos komponentų abstrakciją ir vieną kartą parašius programą, ja bus galima naudotis visose platformose.

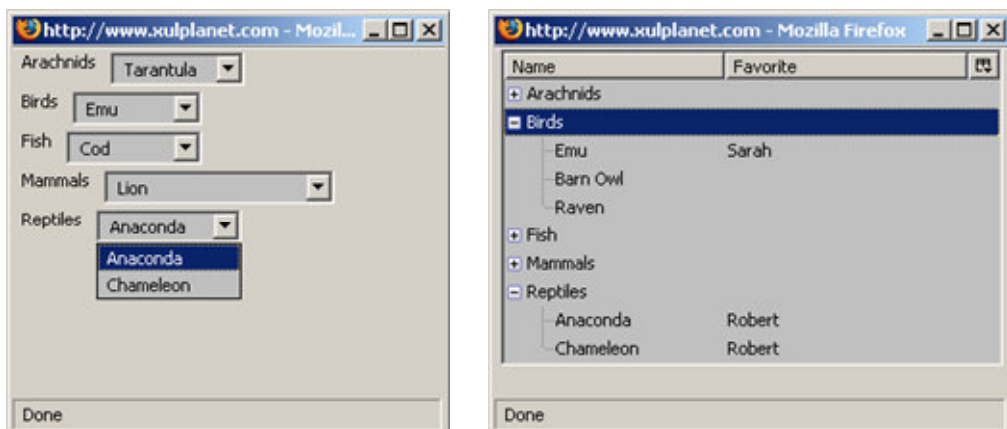
✓ **Programos pateikimo ir programos logikos atskyrimas**

XUL griežtai atskiria kliento programos aprašymą nuo programinės logikos (turinio susidedančio iš XUL, XBL ir Javascript), pateikimą (sąsają susidedančią iš CSS ir paveikslukų) ir nuo kalbos priklausančius teksto žymenis. XUL programų išdėstymas ir išvaizda gali būti keičiama nepriklausomai nuo programos apibrėžimo ir logikos. Toks programos lygmenų atskyrimas sąlygoja lengvesnį programų palaikymą, projektuotojai ir kalbos vertėjai gali lengvai daryti pakeitimus.

✓ **Lengvas modifikavimas, lokalizacija**

Kita praktinė tokio atskyrimo nauda yra ta, kad pateikimas ir kalbos tekstas gali būti lengvai modifikuojamas skirtingiems vartotojams ar vartotojų grupėms. Programuotojai gali turėti vieną programos kodo bazę ir tik modifikuoti įmonės pavadinimą, vartotojo sąsają kiekvienam klientui atskirai.

Pateikiami XUL kalba sukurtos vartotojo sąsajos pavyzdžiai (3 pav.) [4] ir jų XML aprašai (10A priede).



3 pav. XUL vartotojo sąsajos pavyzdys (kairėje – išsiskleidžiantys laukeliai, dešinėje - medis)

2.3.2 BambooKit GUI

BambooKit GUI teikia XML paremtą, daug savybių turinčią vartotojo sąsają [5]. BambooKit yra įrankis gebantis perskaityti, analizuoti ir atvaizduoti XML formatu parašytus vartotojo sąsajos elementus. Tokia vartotojo sąsaja susideda iš tokių elementų: mygtukų,

teksto, meniu, įvedimo laukelių, spalvų pasirinkimo, sąrašo, lentelių, langų ir kt. Visi šie elementai sukuriama naudojant vien tik XML, nereikia rašyti Java kodo.

BambooKit vartotojo sąsaja yra daugiaplatformė, ji sukurta Java programavimo kalba ir veikia daugelyje operacinių sistemų, tokių kaip Windows, Unix / Linux, OS/2, Mac OS 9/Classic, Windows CE / Pocket PC, Symbian OS.

Šis įrankis užima tik 100 KB disko vietos (Jar failas), yra mažas, parašytas Java kalboje ir todėl tinkamas naudoti mobiliuosiuose įrenginiuose, kurie turi JVM, kaip kad įrenginiai su Windows CE ar Symbian platformomis. Savyje jis talpina įvairius vartotojo sąsajos elementus, XML analizatorių, savo įvykių bei objektų piešimo valdymą. Taigi turint ši nedidelį Jar failą ir XML failus aprašančius grafinę vartotojo sąsają, galima matyti šio įrankio teikiamus privalumus.

Trumpai apžvelgsime kaip XML failuose aprašoma vartotojo sąsaja. Kiekvienas XML failas standartiškai turi prasidėti XML antrašte, o toliau dėstomi grafinės vartotojo sąsajos elementai. Tokios sąsajos aprašymo pavyzdys pateikiamas žemiau:

```
<?xml version="1.0" standalone="yes"?>
<Widget setLabel="Hello World" setBackground="red" setSize="100,100">
  <Button setLabel="Click me" setLocation="10,40"></Button>
  <Edit setLabel="type here" setLocation="10,70"></Edit>
</Widget>
```

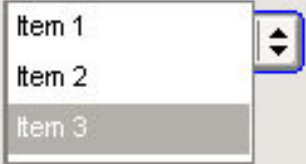
Šis pavyzdys sukuria raudonos spalvos konteinerį su mėlynu mygtuku ir įvedimo laukeliu konteinerio viduje. Dokumentas prasideda XML antrašte, toliau eina pagrindinis elementas savyje turintis visus kitus elementus. Toks aprašymas yra pilnas, teisingai suformuotas XML dokumentas, nes nepažeistos XML sintaksės taisyklės.

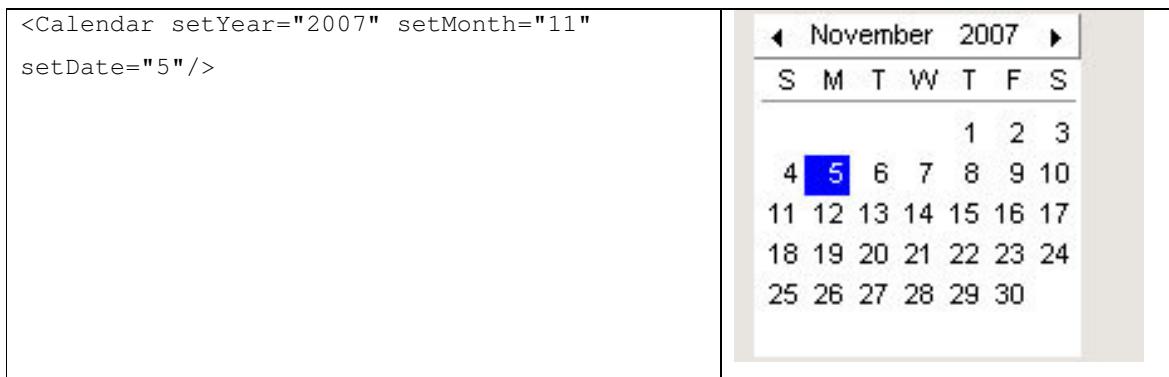
Pagrindiniai elementai, kuriuos gali atvaizduoti Bambookit, yra:

- ✓ Mygtukas
- ✓ Kalendorius
- ✓ Išsiskleidžiantis laukelis
- ✓ Pažymimas laukelis
- ✓ Įvedimo laukelis
- ✓ Sąrašas
- ✓ Meniu
- ✓ Skirtukas
- ✓ Lentelė
- ✓ Langas

Keleto BambooKit grafinių elementų aprašai yra pateikiami (2 lentelė).

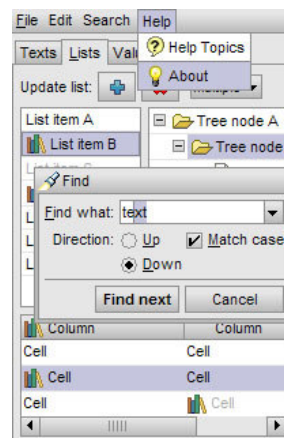
2 lentelė. BambooKit vartotojo sąsajos grafinių elementų aprašai ir pavyzdžiai

Aprašas	Pavyzdys
<pre><Button setLabel="Submit"/></pre>	
<pre><Checkbox setShape="5,5,100,30" setLabel="Checkbox 1"/> <Checkbox setShape="5,40,100,30" setLabel="Checkbox 2"/></pre>	
<pre><Edit setAlphaNumeric="true"/></pre>	
<pre><Combobox setShape="5,5,120,25" setLabel="Select Item"> <Listbox setSize="100,60"> <Item setLabel="Item 1"/> <Item setLabel="Item 2"/> <Item setLabel="Item 3"/> </Listbox> </Combobox></pre>	
<pre><Menu setLabel="Start" setShape="0,220,60,25" setPopupAlign="top"> <Listbox setBorder="flat"> <Item setLabel="Item 1"/> <Item setLabel="Item 2"/> <Menu setLabel="Item 3" setBorder="none" setPopupAlign="right" setPopupRollover="true" setArrow="2"> <Listbox setBorder="flat"> <Item setLabel="Item 1"/> <Item setLabel="Item 2"/> <Item setLabel="Item 3"/> <Item setLabel="Item 4"/> </Listbox> </Menu> <Item setLabel="Item 4"/> </Listbox> </Menu></pre>	



2.3.3 Thinlet

Thinlet yra grafinės vartotojo sąsajos įrankis, kuris nagrinėja grafinės vartotojo sąsajos hierarchiją ir savybes, valdo komunikavimą, atskiria grafinius elementus (apibūdintus XML faile) nuo programos Java programos metodų [6]. Grafinės vartotojo sąsajos pavyzdys pateiktas 4 pav.

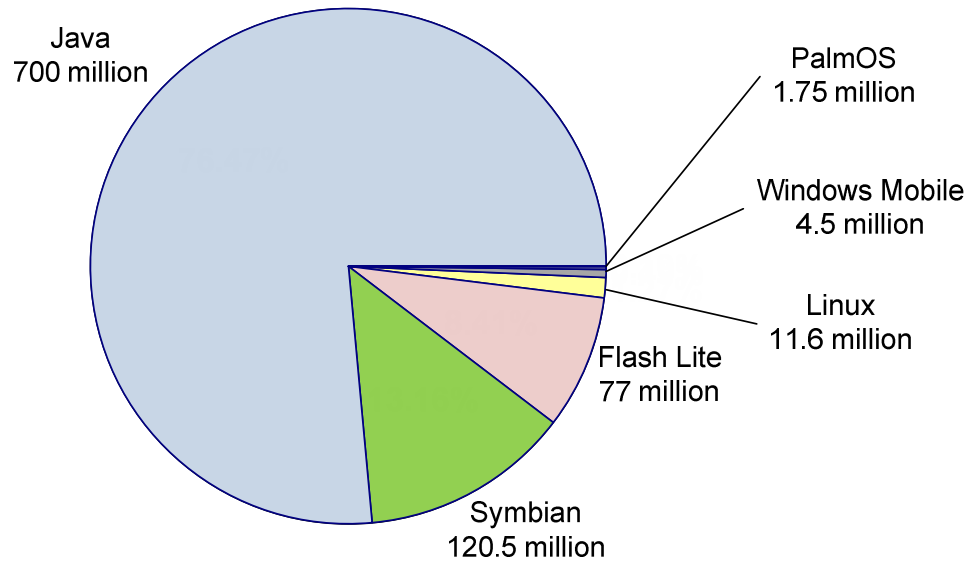


4 pav. Su Thinlet įrankiu sukurtos grafinės vartotojo sąsajos pavyzdys

Thinlet yra palaikomas Java 1.1 - 1.4, Personal Java.

2.4 J2ME pasirinkimo pagrindimas

Viena iš J2ME pasirinkimo priežasčių yra šios kalbos populiarumas. Šiuo metu mobiliųjų įrenginių rinkoje dominuoja Java technologiją palaikantys įrenginiai [7]. Antroje vietoje yra įrenginiai turintys Symbian operacinę sistemą, taip pat gana populiarūs yra įrenginiai su Windows Mobile, Linux platformomis (5 pav.).



5 pav. Mobilųjų įrenginių skirtingų platformų populiarumo statistika

Kita J2ME pasirinkimo priežastis yra ta, kad kitose platformose egzistuoja daugiau ar mažiau panašūs analogai, tačiau jie neturi J2ME palaikymo. O tie kurie turi J2ME palaikomumą, jų funkcionalumas yra labai ribotas.

2.5 J2ME dinaminė vartotojo sąsaja

2.5.1 Vartotojo sąsajos elementai

Mobilieji įrenginiai turi standartinius grafinės vartotojo sąsajos elementus, bei taip pat objektą, kurio pagalba galima piešti įvairias figūras ar elementus.

Pagrindiniai grafinės vartotojo sąsajos elementai yra tokie [8]:

- ✓ Langas
- ✓ Mygtukas
- ✓ Panelis
- ✓ Paveikslukas
- ✓ Įvedimo laukelis
- ✓ Tekstas
- ✓ Bėgantis tekstas
- ✓ Progreso juosta

- ✓ Parinkčių mygtukai
- ✓ Funkcinis mygtukas
- ✓ Sąrašas
- ✓ Lentelė
- ✓ Medis
- ✓ Linija
- ✓ Kvadratas
- ✓ Apskritimas
- ✓ Slinkties juosta

Kiekvienas grafinės vartotojo sąsajos elementas, kuris gali būti pažymėtas, paspaustas ir pan. turi turėti grįžtamąjį ryšį, t.y. jį pažymėjus tai turi atspindėti spalvos pakeitimas, paryškimas, formos pasikeitimas ar žymeklio atsiradimas (6 pav.).



6 pav. Vizualus pažymėto grafinio elemento išsiskyrimas iš kitų

Taip pat elementai turi turėti pavadinimus, kurie identifikuoja elementus. Kiekvienas komponentas gali turėti keletą būsenų, siekiant efektyviai organizuoti grįžtamąjį ryšį su vartotoju. Komponentai, kurių panaudojimas operacijoje nėra leidžiamas ar apibrėžtas, turi būti vartotojui nepasiekiami.

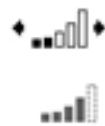
2.5.2 Standartinių komponentų ir Canvas grafinių elementų palyginimas

J2ME turi standartinių objektų rinkinį (7 pav.) [9], kurių funkcionalumas ir savybės yra ribojamos atitinkamų klasių teikiamomis funkcijomis.

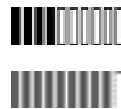


a/ Kalendorius su datos formatu apačioje

b/ Laikrodis su laiko formatu apačioje



c/ Interaktyvus matuoklis



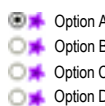
d/ Neinteraktyvus matuoklis



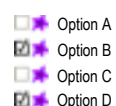
e/ Neapibrėžtas slenkantis matuoklis



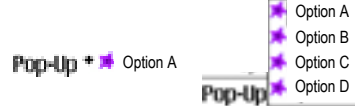
f/ Neapibrėžtas didėjantis matuoklis



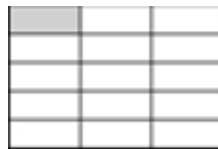
g/ Vieno elemento pasirinkimas



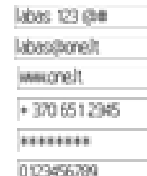
h/ Keleto elementų pasirinkimas



i/ Iššokantis pasirinkimas



j/ Lentelė



k/ Skirtingos paskirties įvedimo laukai (bet kokiems simboliams, el. paštui, nuorodai, tel. nr., slaptažodžiui, skaičiams)



l/ Mygtukas

7 pav. Standartiniai J2ME grafinės vartotojo sąsajos objektai

MIDP vartotojo sąsajos klasės nebuvo rašytos turint omeny jų praplečiamumą, todėl klasės aukšto lygio vartotojo sąsajai kurti nėra paveldimos. Pavyzdžiui, Item klasė, kuri tarnauja kaip bazinė klasė visiems komponentams, kurie gali būti įdėti į formą (Form objektą), nesuteikia priėjimo prie konstruktoriaus, kuris būtų aprašytas kaip public ar protected. Dar svarbiau, nei viena iš aukšto lygio vartotojo sąsajos klasių neleidžia prieiti prie žemo lygio piešimo ar įvykių, kurie yra reikalingi norint rašyti savaip modifikuotus komponentus.

Vienintelis grafinės vartotojo sąsajos praplėtimui tinkamas būdas yra Canvas klasės naudojimas. Ši klasė yra žemo lygio vartotojo sąsajos dalis teikianti būtina informaciją. Tačiau negalima naudoti žemo ir aukšto lygio vartotojo sąsajos komponentų tuo pačiu metu, pavyzdžiui neįmanoma įdėti Canvas objekto į formą, ar įdėti formą į Canvas. Canvas yra praplečiamas ir atsakingas už viso ekrano piešimą, atsakinėjimą visiems gaunamiems

įvykiams. Todėl nusprendus rašyti savo nuosavą klasę, reikia būti atsakingam už visą vartotojo sąsają ir įvykius.

2.5.3 Canvas naudojimo privalumai

Pagrindiniai Canvas klasės naudojimo privalumai yra tokie [10]:

- ✓ Naudojant formas, neretai pastebimos klaidos skirtingų gamintojų mobiliuose įrenginiuose, arba skirtingas tų pačių klasių veikimas. Norint sukurti paprasčiausią meniu, patartina jį nusipiešti pačiam, tada jį galima modifikuoti, kad gerai atrodytų ir kad teisingai pieštų esant skirtingoms ekranų rezoliucijoms. Tuomet galima būti tikram, kad elementai bus nupiešti teisingai.
- ✓ Galima optimizuoti klases ir dėti tik tą kodą, kuris reikalingas atitinkamai programai, vietoj didelių klasių saugojimo, kurių didelė dalis funkcionalumo dažnai yra neišnaudojama.
- ✓ Pastovus priėjimas prie Graphics objekto, kuris leidžia tiesiogiai atnaujinti vaizduojamus elementus ir nereikia laukti kol bus iškviestas `paint()` metodas viso ekrano vaizdo atnaujinimui.
- ✓ Vartotojo modifikuotų bet kokių grafinių elementų kūrimas.

2.5.4 Canvas ir XML

Dinaminės vartotojo sąsajos kūrimas yra paremtas XML kalba [11].

Neigiama XML naudojimo vartotojo sąsajos elementams aprašyti pusė yra ta, kad XML dokumento nagrinėjamas, piešimui reikalingų objektų suradimas, jų savybių skaitymas užtrunka tam tikrą laiko tarpą, kurį vartotojas turi palaukti, kol elementai bus pavaizduoti ekrane.

XML sintaksė turi atitikti standartus [12], visi elementai turi būti uždaryti, t.y. turi turėti žymens pabaigos simbolį, taip pat elementai neturėtų perdengti vienas kito, atributų reikšmės turi būti rašomos kabutėse, naudojant tekstinius simbolius `<` `>` `&` `'` `''`, jie turi būti pakeisti atitikmenimis:

- ✓ `&` (`&`, ženklas)
- ✓ `<` (`<`, mažiau)
- ✓ `>` (`>`, daugiau)
- ✓ `'` (`'`, apostrofa)

- ✓ " (", kabutė)

2.5.5 XML skaitymas ir skaitymo greitis

Duomenų saugojimas XML formatu turi nemažai pranašumų:

- ✓ Duomenys yra lengvai suprantami.
- ✓ Juos lengva susieti su kitais kliento ir serverio duomenimis.
- ✓ Užtikrina, kad duomenys yra teisingi.

Yra sukurta daug XML vertėjų tipų, tačiau daugelis jų naudoja nemažai įrenginio resursų. Yra trys pagrindiniai tipai [13], besiskiriantys savo funkcionalumu:

- ✓ Model vertėjas. Skaito visą dokumentą ir saugo atmintyje, todėl naudoja žymiai daugiau atminties nei kiti vertėjai.
- ✓ Push vertėjas. Skaito visą dokumentą ir praneša listener objektui apie tam tikras dokumento dalis.
- ✓ Pull vertėjas. Skaito dalį dokumento vienu metu. Programa kiekvienos iteracijos metu perskaito dalį dokumento.

XML vertėjai neretai sudaro labai didelius XML dokumentus, kuriuos galima sumažinti taip padidinant programos vykdymo greitį. Yra trys programos greičio optimizavimo kategorijos:

- ✓ Vykdyto pagerinimas
- ✓ Vartotojo išvalgumo lygis
- ✓ Programos kodo dydis

Vykdyto gerinimas yra susijęs su XML dokumento struktūra. XML dokumentas turi talpinti savyje kuo daugiau naudingos informacijos. Dokumentai, siunčiami į mobilųjį įrenginį (iš serverio) neturi turėti savyje papildomos informacijos, kuri nebus naudojama.

Kitas programinės įrangos patobulinimo būdas – vartotojo patirties didinimas. Vertimas į ir iš XML, kaip ir duomenų perdavimas tinklu, turi turėti savo atskirą giją tam, kad vertimo metu vartotojo sąsaja nesustingtų ir lauktų kol veiksmai baigsis. Taip pat turi būti naudojami pull arba push tipo vertėjai tam, kad nereiktų laukti viso dokumento pervertimo.

Mobiliuose įrenginiuose atminties kiekis yra ribotas, dėl to reikia visuomet stengtis kiek įmanoma sumažinti programos dydį. Programa susideda iš klasių failų, paveiksliukų ir kitokių failų ir jos dydžiui sumažinti gali būti panaudotos tam tikros programos, kurios pasižymi tokiomis savybėmis:

- ✓ Pašalina nenaudojamas klases

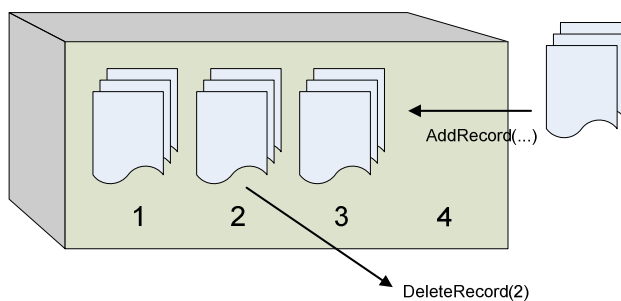
- ✓ Pašalina nenaudojamus metodus ir kintamuosius
- ✓ Pervadina klases, paketus, metodus, kintamuosius
- ✓ Pakeičia klasių failus, kad kompiliatoriai kitaip interpretuotų tam tikras vietas

2.6 Naudojamos technologijos

2.6.1 Record Management System

Mobiliuosiuose įrenginiuose programinė įranga duomenų saugojimui naudoja Record Management System (RMS) dėl mobilių įrenginių ribotų galimybių palyginti su staliniais kompiuteriais. Tokiam duomenų saugojimui Java programavimo kalboje yra specialios klasės, kurios priklauso `javax.microedition.rms` paketui. Kiekvienas įrašas talpina savyje grupę duomenų bazės įrašų [14].

Mobiliųjų įrenginių programos skaito ir rašo įrašus naudodamos unikalų įrašo ID (recordID). Atsiradus naujam įrašui, jam priskiriamas naujas ID. ID intervalas yra nuo 1 iki n, kur n - visų saugojamų įrašų skaičius. Apibendrintai įrašų saugojimas parodytas 8 pav.



8 pav. Įrašų saugojimas įrašų saugykloje mobiliajame įrenginyje

Skirtingi mobilieji įrenginiai turi skirtingo dydžio saugyklas duomenis, todėl svarbu, kad programinėje įrangoje nustatymas duomenų dydžio `MIDlet-Data-Size` atributas neviršytų įrenginio atminties talpos.

`javax.microedition.rms` paketas numato tokias programų vykdymo klaidas [15]:

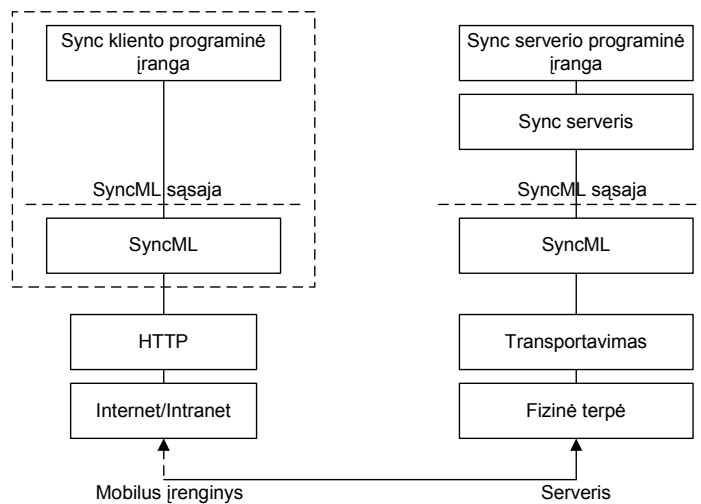
- ✓ `InvalidRecordIDException` – įvyksta, kai įrašo ID yra nesuprantamas.
- ✓ `RecordStoreFullException` – įvyksta, kai įrenginyje pritrūksta laisvos atminties.
- ✓ `RecordStoreNotFoundException` – įvyksta, kai bandoma atidaryti neegzistuojančią duomenų saugyklą.

- ✓ RecordStoreNotOpenException – įvyksta, kai bandoma kreiptis į saugyklą, kuri tuo metu jau būna uždaryta.
- ✓ RecordStoreException – įvykus bet kuriai klaidai iš aukščiau išvardintų.

2.6.2 SyncML protokolas

Norint naudoti nuolat atsinaujinančią informaciją mobiliajame įrenginyje, ją reikia sinchronizuoti su informacija esančia serveryje. Duomenų sinchronizacija veikia taip [16]: mobilus įrenginys vartotojas nepalaiko pastovaus ryšio su interneto tinklu, jis tik parsisiunčia duomenis ir gali su jais dirbti saugodamas juos mobiliajame įrenginyje, o padarius tam tikrus pakeitimus vėl jungiamasi prie serverio ir išsaugomi pakeitimai. Duomenų sinchronizavimo protokolas aprašo duomenų perdavimą sinchronizuojant duomenis, kai įrenginys būna prisijungęs prie tinklo. Protokolas atpažįsta įrašus, turi komandas reikalingas duomenų sinchronizavimui.

SyncML protokolo architektūra [17] matoma 9 pav.



9 pav. SyncML protokolo architektūra

Pagrindiniai SyncML privalumai [18]:

- ✓ SyncML sumažina tinklo vėlinimą, duomenų perdavimo kainą, padidina patikimumą. Duomenų kodavimas, sinchronizavimo komandos, ir pranešimų perdavimas, kur užklausa turi savyje visus padarytus atnaujinimus, o atsakymas pateikia atnaujintą informaciją su iš anksto identifikuotomis klaidomis. Taip pat šis protokolas turi savybę tęsti duomenų perdavimą ir po ilgesniam laikui dingusio ryšio.

- ✓ Palaiko keletą duomenų perdavimo protokolų, tokių kaip HTTP, WAP's Wireless Session Protocol (WSP), Bluetooth's OBEX ir IrDA, SMTP.
 - Palaiko skirtingus duomenų formatus perduodant duomenis
 - Santykiniai duomenys
 - XML ir HTML dokumentai
 - Dvejetainiai duomenys
- ✓ Leidžia pasiekti duomenis iš skirtingų programų – protokolas nepriklauso nuo programavimo kalbos.
- ✓ Atkreipiamas dėmesys ribotiems resursams. Duomenys perduodami dvejetainiam formate siekiant sumažinti atminties poreikį

SyncML Java programavimo kalboje duomenų pranešimams perduoti naudojamas HTTP protokolas, POST komandos siuntimui Javoje naudojama klasė `java.net.HttpURLConnection` [19]. Siunčiamas XML dokumentas turi tam tikrą struktūrą. Informacija esanti tarp žymių `<Alert>` serveriui parodo norimos naudoti lentelės pavadinimą ir sinchronizacijos tipą. `<Cred>` žymės naudojamos autentifikacijai. Autentifikacija gali būti paprasta (basic), ar autentifikacija su MD5 kodavimu, kuri yra saugesnė. `<Meta>` gali būti naudojami daug kur, jų paskirtis yra saugoti papildomai informacijai. `<Last>` ir `<Next>` žymės saugo lentelių įrašų ID. Kliento ir serverio sinchronizacijos pranešimų pavyzdys yra pateiktas 10B priede.

2.7 J2ME testavimo galimybių analizė

2.7.1 Grafinės vartotojo sąsajos testavimas

Programinės įrangos kokybės gerinimas [20] yra svarbus tiek programinės įrangos kūrėjams, tiek vartotojams, todėl programų gamintojams yra svarbu turėti testavimo įrankius. Testavimo įrankiai leidžia kurti testavimo scenarijus, generuoti testavimo atvejus, atlikti vienetų testavimą. Pagrindinė šių įrankių paskirtis yra pagreitinti programinės įrangos kūrimo procesą, aptikti daugiau defektų bei sumažinti programų kūrimo kaštus. Įprastai testavimo scenarijai sudaromi rankiniu būdu arba naudojant įrašymo procedūrą. Plačiai yra naudojamas įrašymo-atkartojimo grafinės sąsajos testavimo būdas. Pirmą kartą testavimas atliekamas rankiniu būdu, testavimo aplinka įrašo testuotojo atliekamus veiksmus grafinėje sąsajoje ir vėliau, atliekant automatizuotą testavimą, šie įvykiai yra atkartojami. Įrašymo-atkartojimo

metodą taikantys testavimo įrankiai nesugeba atpažinti grafinių komponentų, testavimo aplinkos paruošimas reikalauja didelių pastangų

Neretai programinei įrangai testuoti yra naudojamas rankinis testavimas, daugelį kartų vartotojas turi pakartoti tuos pačius veiksmus, aprašytus testavimo atvejuose [21]. Testavimo automatizavimas leidžia ženkliai padidinti testų skaičių. Tuo pačiu padidėjęs testų skaičius nereikalauja didesnių testavimo resursų. Dažniausiai programos funkcionalumas atvaizduojamas ir valdomas iš grafinės vartotojo sąsajos, pažymint skirtingus grafinius elementus, į juos įvedant tekstą, paspaudžiant ir pan., todėl grafinės vartotojo sąsajos testavimas yra labai svarbus.

Šiuo metu mobiliųjų įrenginių rinkoje dominuoja Java technologiją palaikantys įrenginiai [22]. Antroje vietoje yra įrenginiai turintys Symbian operacinę sistemą, taip pat gana populiarūs yra įrenginiai su Windows Mobile, Linux platformomis.

Automatizuoto testavimo atveju vykdomi visi testavimo atvejai, nepriklausomai nuo to, koks buvo vykdymo rezultatas, tokiu būdu garantuojama, kad programos pasikeitimai nepakenkė veikiančioms programos dalims. Scenarijai padeda aptikti programos defektus, jei jiems vykdyti naudojamos skirtingos duomenų aibės, todėl reikia daug pastangų testavimo atvejų bei duomenų parengimui, kad šie galėtų padengti visas programos šakas. Programos kūrimo metu, daugelis kodo vietų keičiamos, todėl reikalingas pakartotinis testų vykdymas. Testavimas yra iteracinis procesas, pagrindiniai iteracijos žingsniai yra testavimo atvejų bei duomenų kūrimas, testų vykdymas ir testavimo rezultatų nustatymas.

Kai kurie testavimo įrankiai interpretuoja programos grafinę sąsają kaip grafinį vaizdą, neišskiriant jame grafinių komponentų. Kiti įrankiai remdamiesi konkrečios technologijos programine sąsaja sąveikauja su grafiniais komponentais, arba programos grafinė sąsaja analizuojama ne tik kaip grafinių komponentų aibė, bet ir kaip sąsaja, kurios pagalba galima kviesti bei vykdyti testuojamos programos funkcionalumą. Šiuo atveju taip pat turėtų būti vykdomas integravimo testavimas. Taip pat neretai yra sekamas ne tik grafinės vartotojo sąsajos kviečiamas bei vykdomas funkcionalumas, taip pat yra atliekamas ir vienetų testavimas.

2.7.2 Testinių atvejų generavimas

Norėdamas sugeneruoti gerus testinius atvejus, testuotojas turi būti įsitikinęs, kad jo testai pilnai padengia programos funkcionalumą [23], bei testai turi pilnai išbandyti pačią grafinę vartotojo sąsają. Norint tai įvykdyti, kyla keletas problemų, tokių kaip: programos dydis, veiksmų seka. Ypatingai tai sukelia sunkumų siekiant atlikti regresinį testavimą.

Skirtingai nuo komandinės eilutės sąsajos, grafinė vartotojo sąsaja turi žymiai daugiau operacijų, kurias reikia ištestuoti. Labai maža programa, tokia kaip Microsoft Word, turi 325 galimas grafinės vartotojo sąsajos operacijas. Didesnėse programose šis skaičius gali žymiai išaugti.

Kita problema yra susijusi su veiksnių seka. Tam tikras sistemos funkcionalumas gali būti pasiektas tik vykdant tam tikrą sudėtingą vartotojo sąsajos įvykių seką. Pavyzdžiui, norėdamas atidaryti failą, vartotojas turi paspausti ant meniu ir tada pasirinkti failo atidarymo operaciją, tada atsidariusiame dialoge įvesti atitinkamą failo vardą ir tada nueiti ant programoje atsidariusio lango. Taigi, didėjantis galimų operacijų skaičius eksponentiškai didina veiksnių sekos problemą. Tai gali būti labai didelė problema, kai vartotojas pats kuria testavimo atvejus.

Grafinės vartotojo sąsajos regresinis testavimas taip pat yra nemaža problema. Vartotojo sąsaja gali žymiai pasikeisti lyginant skirtingas programos versijas. Tam tikra testo veiksnių seka gali neįvykti dėl vieno mygtuko, dialogo, meniu pasikeitimo ar išėmimo.

Yra siūloma daug skirtingų būdų kaip automatizuoti grafinės vartotojo sąsajos testavimą, ar kaip bandyti simuliuoti vartotojo atliekamus veiksmus. Dauguma atvejų stengiasi naudoti komandine eilute paremtas technologijas.

2.7.3 Testinių atvejų vykdymas

Pirmosios testavimo strategijos buvo susietos su komandine eilute. Vienas populiariesnių tokių metodų yra įrašymo/atkartojimo principas. Tai sistema, kai yra įrašomi vartotojo vykdomi veiksmai ekrane ir paskiau vartotojui leidžiama automatiškai atkartoti tuos pačius veiksmus ir palyginti gautus rezultatus su laukiamais.

Įrašymo atkartojimo technologija veikia gan gerai su komandine eilute, bet atsiranda didelių problemų bandant tai įgyvendinti sistemoms turinčioms grafinę vartotojo sąsają [24]. Pati didžiausia problema yra ta, kad ekrane grafinė vartotojo sąsaja gali atrodyti skirtingai programai esant toje pačioje būsenoje. Tai yra dėl to, kad grafinė vartotojo sąsaja leidžia grafiniams objektams keisti išvaizdą ir poziciją ekrane. Šriftai gali būti skirtingi, langų spalvos ir dydžiai gali kisti, bet sistemos duodamas rezultatas gali būti toks pat. Vartotojas tai aiškiai mato, tačiau tai nėra aišku automatinei tikrinimo sistemai.

Siekiant išvengti šių problemų, testuotojai įsimena grafinės vartotojo sąsajos bendravimo duomenis iš langų sistemos. Jie įrašinėja langų įvykius, bendravimas su sistema yra atskiriamas nuo vartotojo sąsajos išvaizdos (įrašinėjama tik įvykių sekos).

Kitas testų vykdymo būdas yra sukurti programos priedą, kuris siųstų programai įvykius ir komandas. Toks tiesioginio įvykių siuntimo ir priėmimo metodas yra naudingas kai

programos įėjimų ir išėjimų testavimas gali būti pilnai automatizuotas ir tokiu būdu išvengiamos vartotojų klaidos.

2.7.4 Mobilųjų įrenginių programų testavimo įrankiai

J2MEUnit

J2MEUnit yra J2ME biblioteka turinti vienetų testavimo karkasą J2ME taikomosios programoms ir yra paremta originaliu JUnit (J2SE) įrankio kodu [25].

Toks karkasas yra reikalingas ir pagrindiniai J2MEUnit ir JUnit skirtumai atsiranda dėl to, kad J2ME neturi objektų atspindžio sąsajos, kaip kad standartinė Java. Todėl J2ME atveju ši trūkumą reikia apeiti modifikuojant kodą, skirtingai įgyvendinant tam tikrus bruožus.

Testavimo atvejų kūrimas yra panašus kaip ir JUnit: sukuriami testiniai metodai J2MEUnit.framework.TestCase poklasėje perdengiant setUp() ir tearDown() metodus, kurie naudojami inicializuoti pradžioje ir sunaikinti kintamuosius ar objektus testų vykdymo pabaigoje. Pagal susitarimą, visi testiniai metodai turi prasidėti žodžiu „test“.

Norint įvykdyti visus testus iš eilės, juos reikia įdėti į testų rinkinį. Žemiau pateiktas testinio atvejo priskyrimo testų rinkiniui pavyzdys:

```
aSuite.addTest(new TestOne("testOne", new TestMethod()  
{ public void run(TestCase tc) {((TestOne) tc).testOne(); } }));
```

Testai gali būti paleidžiami pasinaudojant specialia klase, kuri vadinasi TestRunner ir yra analogiška JUnit atitinkamai klasei. Senesnės J2MEUnit versijos gali veikti tik iš komandinės eilutės, o nuo 1.1 versijos šis įrankis yra patobulintas ir gali veikti emuliaciniame arba mobiliajame įrenginyje.

TestRunner objekto sukūrimas yra gana paprastas ir viskas ką reikia padaryti, tai paduoti testus, kuriuos reikia vykdyti. Tai galima padaryti j2meunit.midletui.TestRunner poklasėje parašius startAdd() metodą ir iškvietus TestRunner start() metodą su testavimo atvejais arba rinkiniais paduotais kaip parametrais:

```
protected void startApp()  
{  
    start(new String[] { "j2meunit.examples.TestAll" });  
}
```

Taip pat, J2MEUnit gali naudoti Antenna įrankius [26]. Ant gali padėti sukurti failus, juo perkopijuoti į reikiamą vietą ir kt.

Sony Ericsson Mobile JUnit

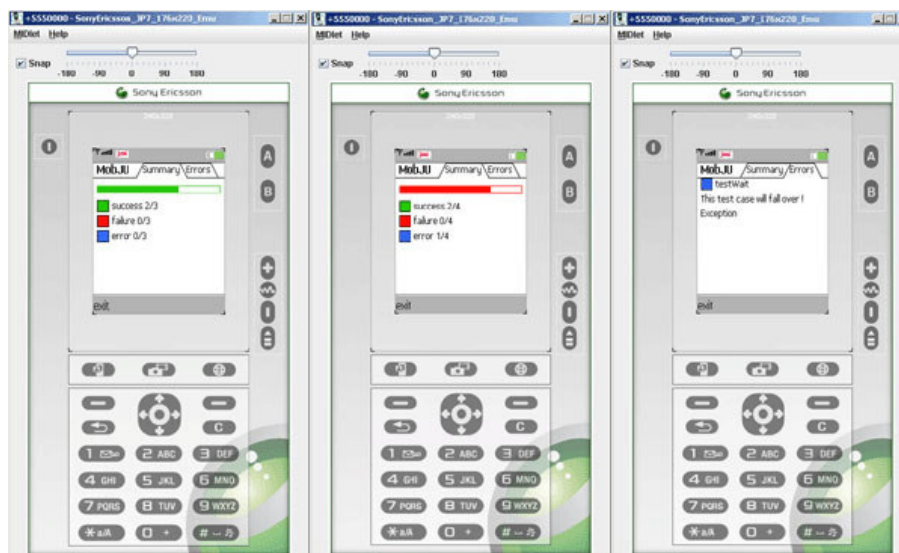
Mobile JUnit yra vienetų testavimo karkasas iš Sony Ericsson, skirtas Java ME CLDC mobiliams telefonams, ir paremtas JUnit atviro kodo karkasu [27].

Vienetų testavimas siekia suskaldyti dideles sistemas į mažesnes dalis ir testuoti kiekvieną dalį užtikrinant teisingą veikimą. Kiekviena dalis turi tam tikrą testinių atvejų skaičių ir įeina į testų rinkinį.

- ✓ Automatinio vienetų testavimo teikiama nauda:
- ✓ Sumažėja kodo nagrinėjimo laikas ir pastangos
- ✓ Saugus kodo pertvarkymas – kadangi yra keičiamos skirtingos sistemos dalys, vienetų testavimas leidžia automatiškai užtikrinti, kad kitos sistemos dalys nepakito.
- ✓ Pakartojamumas – kadangi testai yra automatizuoti, testavimo atvejų pakartojimas yra paprastas
- ✓ Paprasta integracija – vienetų testavimas gali testuoti atskiras dalis ir visas dalis kartu, taip palengvinant integravimo testavimą
- ✓ Dokumentacija – vienetų testavimas teikia naudingus testavimo atvejų pavyzdžius

Paveldėdamas JUnit karkasą, Mobile JUnit leidžia automatizuotą vienetų testavimą mobiliuosiuose telefonuose ir emuliacijoje. Tinklo vėlinimas, prisijungimo kokybė gali neigiamai paveikti testavimo kokybę sudarant vartotojui prastą išpūdį.

Mobile JUnit turi vartotojo sąsają, kurioje rodomas vykdomo testo numeris, bei statistika: kiek testų jau įvykdyta, kiek įvyko be klaidų ir kitą informaciją (10 pav.).



10 pav. Mobile Junit testavimo įrankio vartotojo sąsaja emuliacijoje

Mobile JUnit yra palaikomas visuose Sony Ericsson telefonuose, bet daugumoje kitų telefonų Mobile JUnit taip pat veikia.

2.8 Analizuojamų technologijų pritaikymas

Šiame skyriuje pateikta trumpa technologijų analizė, kuri gali būti panaudota magistro studijų metu sukurtai programinei įrangai patobulinti. Magistro studijų metu sukurta programinė įranga – tai mobilaus vadybininko taikomosios programos, viena dalis yra mobilaus įrenginio programa, kuriai ir bus siūlomi patobulinimai, susiję su grafine vartotojo sąsaja. Platesnis sukurtos programinės įrangos aprašymas yra pateikiamas 3 skyriuje.

3 PROJEKTINĖ DALIS

Šiame skyriuje trumpai apžvelgsime magistratūros studijų metu sukurtos programinės įrangos esminius techninius-projektinius aspektus.

3.1 Sistemos aprašymas

3.1.1 Sukurta sistema

Sukurta programinė įranga – tai mobilaus vadybininko taikomoji programa. Ji leidžia apjungti visas sandėlyje esančias prekes į vieną duomenų bazę, suteikti galimybę vykdyti užsakymus iš mobilios darbo vietos. Atliekamas funkcijas būtų galima skirstyti į keletą grupių: tas, kurios yra atliekamos ar matomos administratoriaus, įmonės vadovo, vadybininko

Mobiliosios technologijos turi nemažų privalumų ir leidžia padidinti įmonės našumą ir konkurencingumą. Įmonei atsiranda galimybė efektyviau išnaudoti darbuotojus, jų darbo vieta yra nepastovi ir jie gali suteikti paslaugas bet kurioje šalies vietoje, nes visa reikalinga informacija yra pasiekama realiu laiku.

Kadangi sistema veikia realiu laiku, todėl užsakymų apdorojimas, sandėlio atsargų apskaita, produkcijos apskaita ir išvežimo kontrolė, produkcijos paskirstymo planavimas, kontrolė yra atliekami iškart, kai tik informacija apie užsakymą pasirodo sistemoje. Taip planuodama bei kontroliuodama savo veiklą įmonė gali daug efektyviau išnaudoti savo resursus.

3.1.2 Funkcionalumas

Vienas pagrindinių tikslų yra mobilaus vadybininko taikomosios programinės įrangos kūrimas ir testavimas. Jos pagalba vadybininkas galės gauti pačią naujausią informaciją apie produktus, užsakymus, bei klientams teikti šias paslaugas:

- ✓ Sukurti naują užsakymą
- ✓ Įtraukti prekes į užsakymą
- ✓ Pašalinti prekes iš užsakymo
- ✓ Įvykdyti užsakymą

- ✓ Peržiūrėti įvykdytus/neįvykdytus užsakymus
- ✓ Peržiūrėti atsargas
- ✓ Generuoti ataskaitas
- ✓ Peržiūrėti ataskaitas
- ✓ Įvesti informaciją apie naują klientą
- ✓ Peržiūrėti informaciją apie klientus
- ✓ Keisti sistemos kalbą
- ✓ Sinchronizuoti duomenis

Ši programinė įranga įgyvendinta Java programavimo kalboje, todėl tinkama platesniam mobiliųjų įrenginių ratui: mobiliam telefonui, kišeniniam kompiuteriui, nešiojamam kompiuteriui. Klientinė dalis GPRS pagalba jungiasi prie serverio ir realiu laiku gali gauti visą reikalingą informaciją.

3.1.3 Adresatas

Kuriamos programinės įrangos adresatas yra didmeninės prekybos kompanijos, kurios užsiima užsakymų vykdymu bei prekių pristatymu klientams esantiems daugiau ar mažiau pasiskirsčiusiems skirtingose šalies ar net pasaulio vietose. Programinė įranga ypač naudinga ir padidina įmonės darbo efektyvumą, kai yra didelis klientų kiekis.

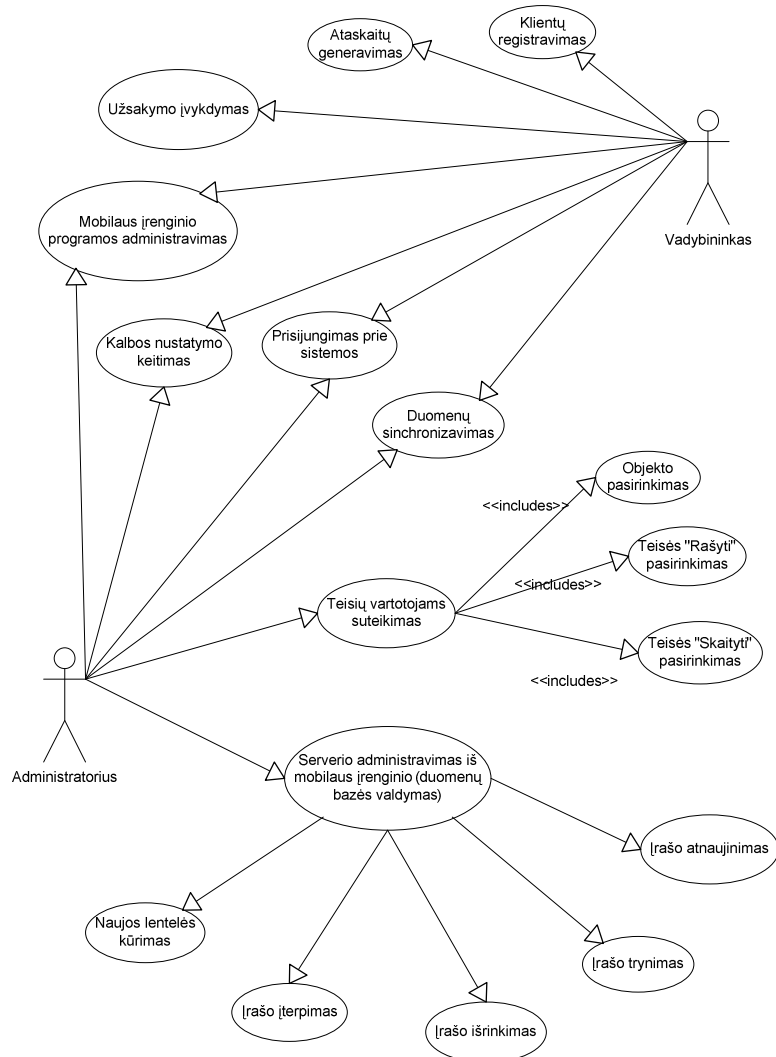
Pagrindiniai programinės įrangos teikiami privalumai:

- ✓ Žymiai supaprastėja užsakymų apskaita:
 - Užsakymų peržiūra
 - Užsakymų įvykdymas
 - Tikslesnės ir greitesnės sumos apskaičiavimo, prekių nurašymo iš sandėlio operacijos
- ✓ Lengviau paskirstyti klientus darbuotojams
- ✓ Nereikia papildomo personalo apskaitai, dokumentų pildymui
- ✓ Atkrenta papildomas darbas statistikos rinkimas metinėms ar kito laikotarpio ataskaitoms
- ✓ Geresnė darbuotojų kontrolė

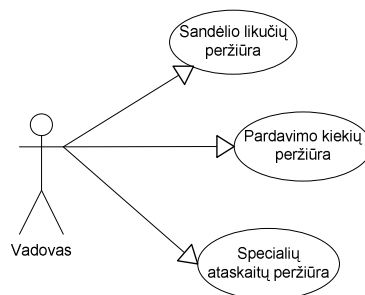
3.2 Sistemos realizacija

3.2.1 Panaudojimo atvejai

11 pav. ir 12 pav. pavaizduoti pagrindiniai mobilus vadybininko programinės įrangos panaudos atvejai.



11 pav. Vadybininko ir administratoriaus programinės įrangos panaudos atvejai



12 pav. Vadovo programinės įrangos panaudos atvejai

Sistemos vartotojai yra šie:

- ✓ Vadovas
- ✓ Administratorius
- ✓ Vadybininkas

Pagrindinių panaudos atvejų aprašai:

- ✓ Klientų registravimas
Užregistravus vartotoją bus kaupiama visa pardavimų apie jį informacija.
- ✓ Ataskaitų generavimas
Vadybininkas pagal pasirinkimą galės peržiūrėti savo tam tikro laikotarpio pardavimus, arba atskiro kliento pirkimus per tam tikrą laikotarpį.
- ✓ Kalbos nustatymo keitimas
Vartotojas kiekvienas asmeniškai galės nusistatyti sistemos kalbą. Pakeitus kalbą visos programos vartotojo sąsajos kalba pasikeis, įtraukiant visus meniu punktus, ataskaitas.
- ✓ Mobilaus įrenginio programos administravimas
Vartotojui bus leidžiama pasirinkti kaip jis nori matyti informaciją, t.y. išdėstymą, pateikimo formą.
- ✓ Prisijungimas prie sistemos
Kiekvienai vartotojų grupei bus atskiras prisijungimo langas. Įvedus teisingą prisijungimo vardą ir slaptažodį, vartotojas galės naudotis jam prieinamomis programos funkcijomis.
- ✓ Duomenų sinchronizavimas
Po padarytų tam tikrų pakeitimų, vartotojas galės sinchronizuoti duomenis, t.y. iš mobilaus įrenginio atnaujinti juos serveryje esančioje duomenų bazėje.
- ✓ Vartotojų teisių suteikimas
Šios administratoriaus funkcijos pagalba galima suteikti teises programinės įrangos vartotojams. Uždraudus rašymą, vartotojas negalės išsaugoti padarytų pakeitimų, o uždraudus skaitymą, informacija nebus rodoma apskritai.
- ✓ Serverio administravimas iš mobilaus įrenginio (duomenų bazės valdymas)
Administratoriui suteikta galimybė prisijungti prie serverio iš mobilaus įrenginio ir redaguoti duomenis: įrašyti, trinti, peržiūrėti, atnaujinti įrašus, kurti naujas lenteles. Į tai įeis taip pat ir vadybininkų prisijungimų, prekių ir kitos informacijos sukūrimas, trynimasis, redagavimas.

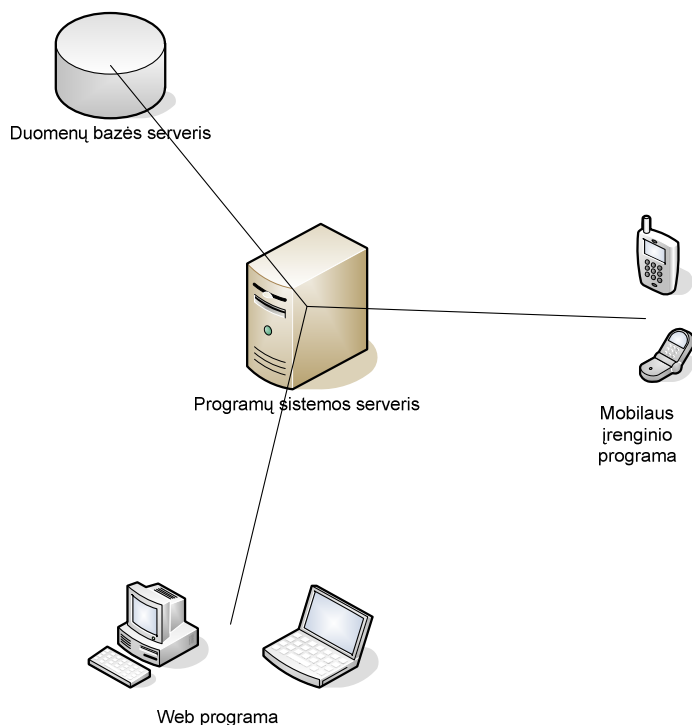
Vadovas galės naudotis daugeliu administratorius ir dar keletu papildomų funkcijų.

- ✓ Sandėlio likučių, pardavimo kiekių, specialių ataskaitų peržiūra
- Tai galės daryti tik įmonės vadovas, kuris pagal norą ir pasirinkimą galės peržiūrėti visas ataskaitas.

3.3 Architektūra

3.3.1 Apžvalga

Programinė įranga skirta įmonėms, kurios intensyviai bendrauja su savo klientais, atlieka užsakymus nutolus, pas klientus. Sistemos bendras vaizdas yra pateiktas 13 pav.



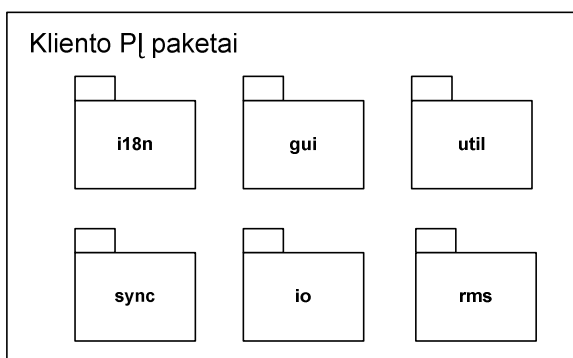
13 pav. Bendras sistemos išsidėstymas

Sistema susideda iš šių pagrindinių dalių:

- ✓ Mobilus įrenginio programinė įranga
- ✓ Serverio programinė įranga
- ✓ Web sąsaja
- ✓ Duomenų bazė

3.3.2 Sistemos statinis vaizdas

Bendras sistemos paketų vaizdas parodytas 14 pav. ir 15 pav. Klientinė programos dalis esanti mobiliajame įrenginyje, o serverinė dalis vykdo visas reikalingas operacijas su duomenų bazės įrašais. Šias programos dalis sudaro paveikslėliuose parodyti paketai. Serverinės dalies microedition paketas susideda iš trijų paketų: io, rms ir sync.



14 pav. Mobilus įrenginio programos paketai

Mobiliojo įrenginio sistemos kalbos paketas (i18n) leidžia pakeisti sistemos kalbą, kalbos duomenys saugojami XML failuose. Nesunkiai galima įdėti naujas kalbas, sukūrus naujų kalbų duomenų XML failus ir įkėlus į atitinkamą direktoriją.

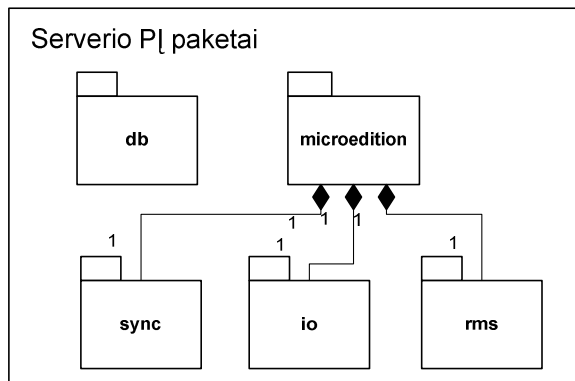
Grafinės vartotojo sąsajos paketas (gui) apibrėžia vartotojo sąsajos elementus, prisijungimo langą, formas ir dialogus. Jais naudojasi visi sistemos vartotojai

Klientinės programinės įrangos dalies konfigūracinis paketas (util) apima įvairius sistemos nustatymus, tokius, kaip kalbos, IP adreso, kodavimo ir kiti nustatymai

Duomenų sinchronizacijos paketas (sync) naudoja SyncML duomenų perdavimo protokolą, taip perduodami duomenys tarp mobilus įrenginio ir serverio.

Paketas informacijos saugojimui failuose, duomenų siuntimui (io). Ryšio sudarymas reikalingas, kad duomenys būtų galima perduoti į serverį ir atlikti atitinkamus veiksmus, bei gauti duomenis iš serverio.

Paketas duomenų saugojimui mobiliajame įrenginyje (rms). Tam tikras kiekis įrašų gali būti išsaugojami mobilus įrenginio atmintyje iki tam tikro laiko kol jie bus sinchronizuojami su serveriu.



15 pav. Serverio programinės įrangos paketai

Duomenų bazės paketas (db) – tai serverinės dalies paketas kur sudaromas prisijungimas prie duomenų bazės, saugoma informacija apie duomenų bazę bei generuojamos SQL užklauskos

RMS įrašų valdymo sistemos paketas (rms) apibrėžia duomenų saugojimo struktūrą, lenteles, operacijas, kurias galima atlikti su duomenų įrašais.

Duomenų sinchronizavimo paketas (sync) - serverio programinės įrangos dalies SyncML protokolo įgyvendinimo paketas. SyncML protokolas atsakingas už duomenų mainus tarp mobilaus įrenginio ir serverio programų

3.4 Programų sistemos diegimas bei administravimas

3.4.1 Reikalinga techninė ir programinė įranga

Sistemai reikalinga techninė įranga:

✓ **Mobilus įrenginys**

Mobiliajam įrenginiui pagrindiniai reikalavimai yra tokie:

- Internetas: Wi-Fi ir/arba GPRS
- J2ME palaikymas.
- Min 128 Kb JAR atmintis

GPRS technologijos pagalba, mobilus ryšys gali jungtis prie sistemos serverio. GPRS paslaugą tiekia mobilaus ryšio tiekėjai. Susisiekite su Jūsų mobilių paslaugų tiekėjų arba aplankykite jų interneto svetaines, kur Jūs rasite daugiau informacijos apie šios paslaugos įjungimą.

✓ **J2EE serveris**

Aptarnauja klientų užklausimus ir tarpininkauja jiems dirbant su duomenų baze. Kadangi serverio sistemos pusė turi būti nepriklausoma nuo platformos, ji bus realizuota J2EE platformoje. Kompiuteryje turi būti įdiegta Java Runtime Environment 1.5.

Pagrindiniai reikalavimai:

- Serverio procesorius 800 MHz
- RAM atmintis Min 256 MB
- 1,5 Gb laisvos vietos diske
- Interneto ryšys

✓ **Duomenų bazės serveris**

Duomenų bazė diegiama Microsoft Windows XP operacinėje sistemoje.

Duomenų bazės valdymo sistemai naudojama IBM WorkBench.

Pagrindiniai reikalavimai:

- Serverio procesorius 500 MHz
- RAM atmintis Min 128 MB
- 10 Gb laisvos vietos diske
- Interneto ryšys

✓ **Kliento stalinis kompiuteris Web sąsajai**

Reikalingas Web sąsajai, gali būti bet kokia operacinė sistema. Kompiuteris turi būti prijungtas prie Interneto. Reikalavimai yra minimalūs, tokie, kad veiktų bet kuri grafinė Interneto naršyklė, bei Interneto ryšys.

3.4.2 Pagrindinės problemos

Pagrindinės iškylančios ar galimos problemos yra šios:

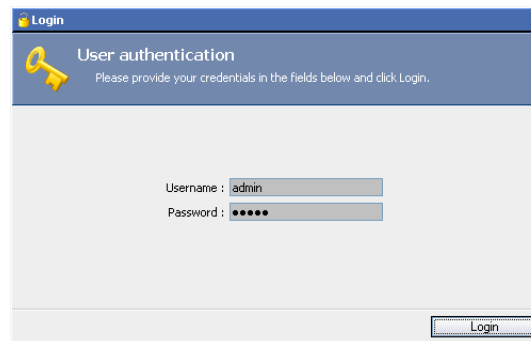
- ✓ Mobilaus įrenginio ar kitos techninės įrangos gedimai
- ✓ Mobilaus įrenginio ribota atmintis
- ✓ Mobilaus įrenginio išsikrovimas – tada gali būti prarasti tam tikri dar neišsaugoti duomenys. Įrenginius reikia visuomet laikyti pakrautus, nes išsikrovus, priklausomai nuo įrenginio tipo, Jūs galite prarasti įrenginio atmintyje saugomą informaciją. Vadybininkų automobiliuose turėtų būti įrengta galimybė įrenginius pasikrauti.
- ✓ Interneto ryšio nebuvimas
- ✓ Neretai lėtas ryšys

3.4.3 Sistemos administravimo Web sąsaja

Sistemos administravimas vyksta per Web sąsają, naudojantis nuoroda:

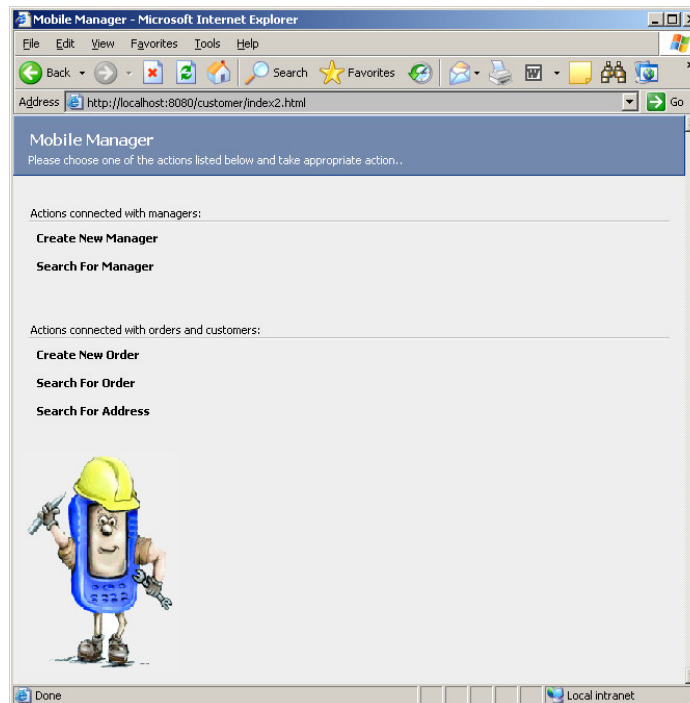
<http://localhost:8080/customer/>

Administratoriui pirmiausia parodomas prisijungimo langas, kur prisijungimui reikalingas vartotojo vardas ir slaptažodis (Prisijungimo vardas: admin, slaptažodis: admin). Prisijungimo langas matomas 16 pav.



16 pav. Web sąsajos administratoriaus prisijungimo langas

Administratoriui prisijungus prie sistemos, parodomas visų pagrindinių funkcijų sąrašas, kaip parodyta 17 pav.



17 pav. Web sąsajos pagrindinis administravimo langas

3.5 Sistemos patobulinimo apžvalga

Šiame skyriuje aprašyta magistro studijų metu sukurta programinė įranga (mobilaus vadybininko taikomosios programos). Viena jos dalis - mobilaus įrenginio programa gali būti patobulinta vartotojo sąsajos atžvilgiu, suteikiant jai daugiau funkcionalumo, atitinkamą išvaizdą ir patogią navigaciją.

Todėl vienas pagrindinių darbo tikslų yra vartotojo sąsajos tyrimas ir universalių grafinės sąsajos kūrimo technologijų pasiūlymai, kurios pagalba vietoj standartinių J2ME komponentų būtų kuriami savi komponentai, turintys praplėstą funkcionalumą ir išvaizdą bei aprašomi XML dokumente. Platesnis tyrimo ir galimų sprendimų aprašymas pateikiamas 4 skyriuje.

4 TYRIMO DALIS

Šiame skyriuje apžvelgsime siūlomą dinaminės vartotojo sąsajos mobiliems įrenginiams kūrimo metodą.

Magistratūros studijų metu buvo vykdomas programinės įrangos mobiliems įrenginiams kūrimo projektas. Projekto pasėkoje buvo numatytos programų kūrimo procesą gerinančios galimybės. Šių galimybių realizavimui siūlomas dinaminės vartotojo sąsajos mobiliems įrenginiams kūrimo metodas.

4.1 Tikslai

Vienas pagrindinių tikslų yra dinaminės vartotojo sąsajos kūrimas mobiliosioms platformoms, suteikiant priėjimą prie beveik visų mobilaus įrenginio teikiamų funkcijų. Pagrindiniai dinaminei programinei įrangai vartotojo sąsajai keliami reikalavimai yra šie:

- ✓ Žemo lygio taikomosios programos sukūrimas, kuri skaitytų ir nagrinėtų XML failų turinį ir pagal tai kurtų vartotojo sąsają
- ✓ Lango ir kitų elementų aprašymas XML kalba, perėjimas tarp skirtingų langų, grįžimas į ankstesnijį
- ✓ Palyginti paprastos ir dinaminės vartotojo sąsajos vėlinimą, t.y. laiko trukmę nuo vartotojo veiksmo pradžios iki atvaizduoto rezultato (pavyzdžiui perėjimas iš vieno lango į kitą)
- ✓ Atlikti XML failų skaitymo ir nagrinėjo greičio testus, siūlyti būdus proceso pagreitinimui
- ✓ Remiantis ta pačia XML paremta programos architektūra, atlikti galimų testavimo būdų tyrimą

4.2 Problemos

Programavimo technologijos mobiliems įrenginiams nėra orientuotos į konkrečią aparatūrinę architektūrą ir gali būti taikomos įvairiems mobiliems įrenginiams, palaikantiems J2ME technologiją. Mobilųjų technologijų standartai neapibrėžia kaip aparatūros gamintojai turėtų realizuoti technologijos palaikymą. Todėl grafinė vartotojo sąsaja skirtingų gamintojų įrenginiuose skiriasi viena nuo kitos. Kuriant vartotojo sąsają reikia atsižvelgti į įrenginių

skirtumus, tokius kaip skirtingas ekrano dydis ar skirtingi mygtukai naudojami tam pačiam funkcionalumui pasiekti. Todėl vartotojo sąsaja turi būti išbandyta visuose įrenginiuose, į kuriuos yra orientuota programa.

Grafinės vartotojo sąsajos ir programos operacijų scenarijų testavimas mobiliems įrenginiams dažniausiai yra atliekamas rankiniu būdu. Testavimą galima automatizuoti iš dalies, tačiau visiškas testavimo automatizavimas mobiliųjų įrenginių vartotojo sąsajai yra praktiškai neįmanomas.

Rinkoje egzistuoja labai didelė mobiliųjų įrenginių įvairovė ir kiekvienas įrenginys skiriasi savo išvaizda, funkcinėmis galimybėmis, ekrano dydžiu ir daugeliu kitų savybių.

Įrenginio funkcionalumas bei savybės neretai priklauso nuo įrenginio paskirties. Pavyzdžiui, įrenginiai skirti rašyti dokumentams bei el. laiškamams turi didesnę ekrano rezoliuciją ir patogesnę klavišų išsidėstymą.

4.3 Sprendimo būdų apžvalga

J2ME turi Canvas objektą, kuris leidžia piešti bet kokias figūras ir elementus bei palaiko įvykius, tokius kaip mygtukų paspaudimai. Canvas pagalba galima piešti bet kokius objektus, pavaizduoti paveikslukus ir tekstą, t.y. kurti nepriklausomą vartotojo sąsają.

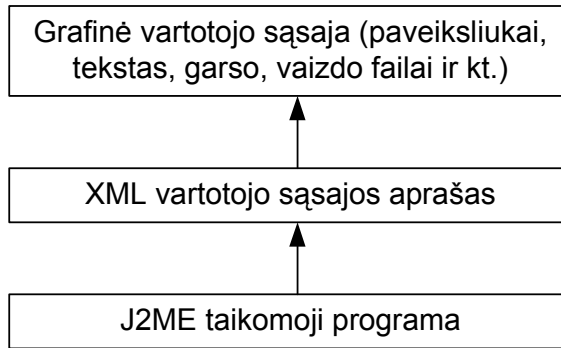
Dinaminei vartotojo sąsajai aprašyti gali būti naudojama XML kalba, kur vienas XML failas atitinka vieno vaizduojamo lango vaizdą. J2ME programa galėtų skaityti tokius failus ir pagal jų turinį atvaizduoti grafinius elementus ir teikti atitinkamą funkcionalumą. Kiekvieno XML failo turinys yra apgaubiamas `<window></window>` žymomis, kurios identifikuoja, kad tai yra atskiras lango elementas. XML dokumentas apdorojamas XML analizatoriumi, atėjus į puslapį, visas XML failas perskaitomas į atmintį. Toliau perskaityti duomenys naudojami elementų išdėstymui ir atvaizdavimui. Kiekvienam langui naudojamas atskiras failas (pavyzdžiui `page1.xml`, `page2.xml` ir tt.).

Taip pat būtina tokios programos veikimo greičio patikra: laiko trukmė per kurią programa reaguoja į vartotojo veiksmus (skaito, nagrinėja XML failus, atvaizduoja grafinius elementus). Perėjimas tarp langų neturėtų užtrukti ilgiau nei 1-2 sekundes. Taip pat lange esant dideliame elementų kiekiui, sąsaja neturėtų labai žymiai sulėtėti.

4.4 Sistemos struktūra

4.4.1 Bendras vaizdas

Sistema susideda iš trijų lygių (18 pav.), kurių žemiausiasis yra J2ME programa, ji skaito XML dokumentą, jį nagrinėja ir pagal tai kuria vartotojo sąsają. Apie J2ME programos veikimą bei architektūrinius sprendimus daugiau aprašyta 4.4.2 skyriuje.

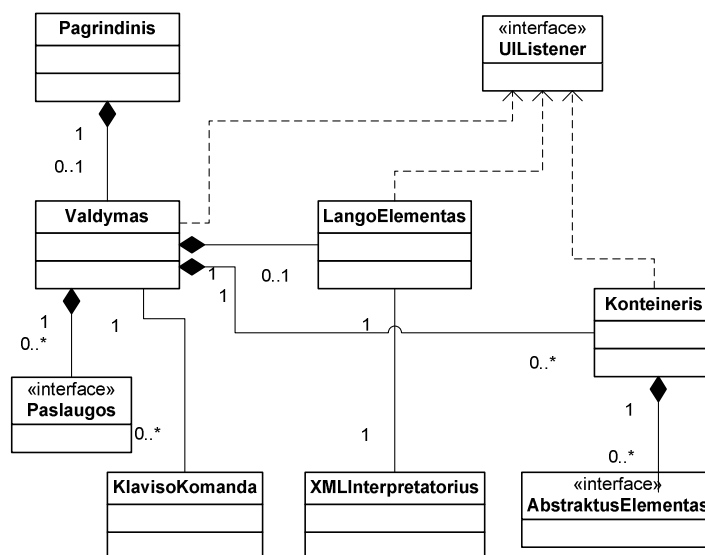


18 pav. Dinaminės vartotojo sąsajos sistemos struktūra

Kiekvienas grafinės vartotojo sąsajos aprašo XML dokumentas susideda iš pagrindinio – lango elemento ir jame esančių kitų elementų. Šiuose XML dokumentuose gali būti nuorodos į naudojamus paveikslukų, tekstų, garso ar vaizdo failus. Detalesnis XML kalba aprašomų elementų apibūdinimas pateiktas 4.5 skyriuje.

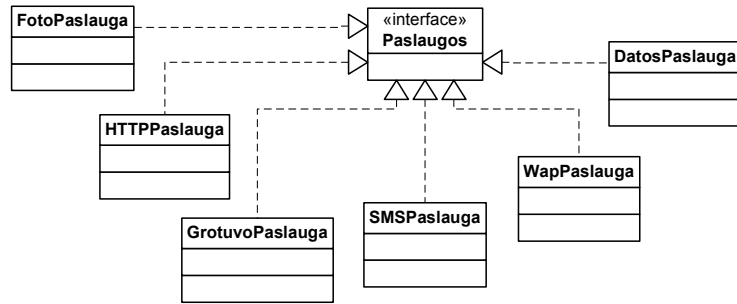
4.4.2 J2ME taikomosios programos architektūriniai sprendimai

Šiame skyriuje pateikia J2ME taikomosios programos detalesnis vaizdas (19 pav.).



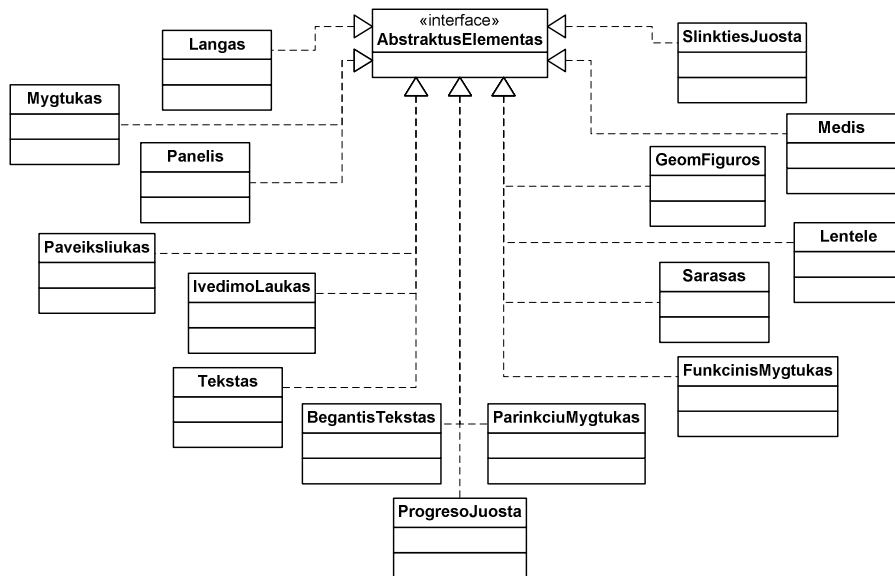
19 pav. J2ME taikomosios programos bendras architektūrinis vaizdas

Iš pateikto vaizdo galime išskirti keletą objektų ir juos detaliau apibūdinti. Vienas iš jų - Paslaugos (20 pav.). Visos paslaugos turi bendrą sąsajos klasę. J2ME visos paslaugų klasės yra pasiekiamos vartotojams per XML kalboje aprašytas funkcijas, kurios detaliau aprašytos 4.6 skyriuje.



20 pav. J2ME taikomosios programos paslaugų diagrama

Kitas bendras sąsajos objektas yra abstraktus elemento klasė (21 pav.). Ji susieja visų elementų bendrus atributus, funkcijas, o kiekviena elemento klasė atskirai gali turėti pagal savo funkcionalumą praplėstą atributų ar funkcijų skaičių.



21 pav. J2ME taikomosios programos elementų diagrama

4.5 Dinaminės vartotojo sąsajos elementų aprašymas

Šiame skyriuje pateikiamas dinaminės vartotojo sąsajos aprašymui XML kalba naudojami elementai.

4.5.1 Abstraktus grafinis elementas, elementų funkcionalumas

Visi mobilieji įrenginiai turi standartinius grafinės vartotojo sąsajos komponentus, bei taip pat komponentą, kurio pagalba galima piešti įvairias figūras ar elementus. Toks komponentas J2ME kalboje yra Canvas objektas, kurio pagalba galima kurti vartotojo sąsają. Šiame skyriuje apibrėšime pagrindinius kuriamus (piešiamus) vartotojo sąsajos elementus ir juos visus siejančias savybes bei funkcionalumą.

Pagrindiniai grafinės vartotojo sąsajos elementai bus tokie:

- ✓ Langas
- ✓ Mygtukas
- ✓ Panelis
- ✓ Paveikslukas
- ✓ Įvedimo laukelis
- ✓ Tekstas
- ✓ Bėgantis tekstas
- ✓ Progreso juosta
- ✓ Parinkčių mygtukai
- ✓ Funkcinis mygtukas
- ✓ Sąrašas
- ✓ Lentelė
- ✓ Medis
- ✓ Linija
- ✓ Kvadratas
- ✓ Apskritimas
- ✓ Slinkties juosta

Kiekvienas grafinės vartotojo sąsajos elementas, kuris gali būti pažymėtas, paspaustas ir pan. turi turėti grįžtamąjį ryšį, t.y. jį pažymėjus tai turi atspindėti spalvos pakeitimas, paryškimas, formos pasikeitimas ar žymeklio atsiradimas. Taip pat elementai turi turėti pavadinimus, kurie reikalingi vėlesniam komponentu panaudojimui ar jų savybių keitimui. Kiekvienas komponentas gali turėti keletą būsenų, siekiant efektyviai organizuoti grįžtamąjį ryšį su vartotoju. Komponentai, kurių panaudojimas operacijoje nėra leidžiamas ar apibrėžtas, turi būti vartotojui nepasiekiami.

Yra keletas pagrindinių savybių, prieinamų vartotojui iš XML dokumento, kurias turi visi be išimties elementai (3 lentelė). Kadangi šiuos atributus turi visi elementai, jie dažnai nebus minimi prie atskirų elementų aprašymo, o apibūdinami tik tie atributai, kurių funkcionalumas skirtinguose elementuose yra skirtingas arba be jų elementas negali normaliai funkcionuoti.

3 lentelė. Grafinio vartotojo sąsajos abstraktaus elemento atributai ir jų aprašymas

Savybė	Paskirtis
name	Elemento ID
x	Elemento pozicija X ašies atžvilgiu skaičiuojant nuo viršutinio kairiojo ekrano kampo
y	Elemento pozicija Y ašies atžvilgiu skaičiuojant nuo viršutinio kairiojo ekrano kampo
w	Plotis
h	Aukštis
color	Elemento teksto spalva
selectedcolor	Pažymėto elemento teksto spalva
backgroundcolor	Elemento fono spalva
selectedbackcolor	Pažymėto elemento fono spalva
bordercolor	Apvado spalva
selectedbordercolor	Pažymėto apvado spalva
backgroundimage	Elemento fono paveikslukas
selectedbackimage	Pažymėto elemento fono paveikslukas
visible	Nusako ar elementas yra matomas, ar paslėptas
showborder	Nusako ar apvadas yra matomas
text	Elemento vaizduojamas tekstas, jei elementas jį turi
fonttype	Šrifto tipas
fontsize	Šrifto dydis
fontstyle	Šrifto stilius
onclick	Veiksmai vykdomi paspaudus elementą

Svarbiausi elementai detaliau apžvelgiami sekančiuose poskyriuose.

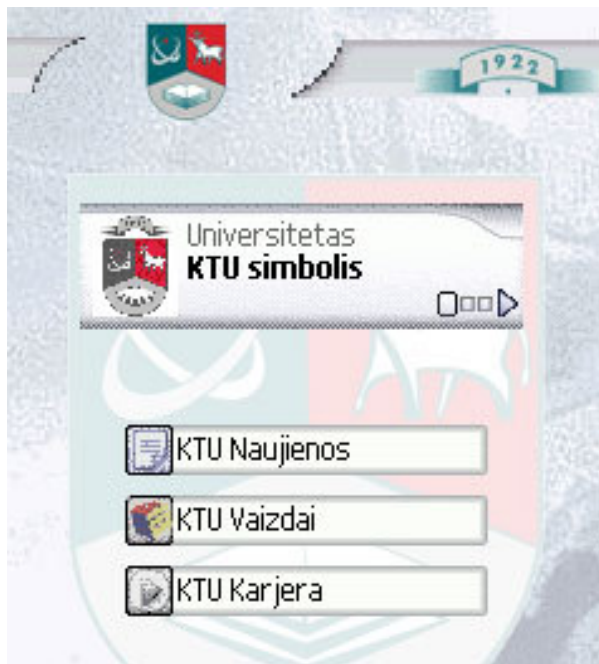
4.5.2 Lango

Lango yra visą ekraną apimantis elementas, jis turi jam priskirtus veiksmus ir savybes, kurie apsprendžia jo išvaizdą ar elgseną. Kokie veiksmai turėtų būti atlikti lango užkrovimo metu ir kitas savybes apibrėžia lango atributai (4 lentelė).

4 lentelė. Lango elemento atributai ir jų aprašymas

Savybė	Paskirtis
onload	Veiksmai vykdomi lango užsikrovimo metu
backgroundimage	Viso lango fono paveikslukas
title	Lango pavadinimas

22 pav. parodytas lango pavyzdys, kuriame yra įdėta daugelis kitų elementų: paveikslukai, mygtukai, tekstas ir kt.



22 pav. Lango elementas su kitais elementais jo viduje

Lango aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<window w="176" h="208" backgroundimage="background.jpg"
onload="goto('next.xml');"> ... </window>
```

4.5.3 Mygtukas

Mygtukas yra fokusuojamas elementas, kuris turi jam priskirtus veiksmus, kurie atliekami jo sufokusavimo ar paspaudimo metu. Kokie veiksmai turėtų būti atlikti mygtuko paspaudimo metu ir kitas savybes apibrėžia mygtuko atributai (5 lentelė).

5 lentelė. Mygtuko elemento atributai ir jų aprašymas

Savybė	Paskirtis
onfocus	Veiksmai vykdomi elemento sufokusavimo metu
focused	Nustato ar elementas yra sufokusuotas. Jei šis atributas nėra nustatymas nei vienam puslapio elementui, pagal nutylėjimą bus sufokusuotas pirmas puslapio fokusuojamas elementas
focusable	Nustato ar elementas apskritai gali būti fokusuojamas
align	Horizontalus mygtuko teksto išlygiavimas

23 pav. parodytas mygtukų pavyzdys, pirmasis yra pažymėtas, tai matosi vizualiai nes jis išsiskiria apvado ir teksto spalva.



23 pav. Mygtukų pavyzdys (pirmasis yra pažymėtas)

Mygtuko aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<button name="button1" x="50" y="50" backgroundimage="buttonback.png" selectedbackimage="buttonselected.png" text="Mygtukas" onclick="goto('next.xml');"/>
```

4.5.4 Panelis

Šis elementas gali būti fokusuojamas ir turėti savyje daugelį kitų elementų. Jis taip pat turi jam priskirtus atributus ir veiksmus, kurie atliekami jo sufokusavimo ar paspaudimo metu. Kokie veiksmai turėtų būti atlikti panelio paspaudimo ar fokusavimo metu ir kitas savybes apibrėžia panelio atributai (6 lentelė).

6 lentelė. Panelio elemento atributai ir jų aprašymas

Savybė	Paskirtis
selectedbackimage	Pažymėto panelio fono paveikslukas
textcolor	Elemento teksto spalvos šešiolyktainė reikšmė
focusable	Nustato ar elementas gali būti fokusuojamas

Panelis išsiskiria iš kitų elementų tuo, kad jis gali savyje turėti kitus elementus, kurie rašomi tarp panelio pradžios ir pabaigos žymenų. Jei elementų yra daug, jis gali turėti slinkties juostą. Kai yra daugiau konteinerių, pagal nutylėjimą bus sufokusuotas pirmasis lange esantis fokusuojamas panelis.

Panelio aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<panel x="15" y="15" w="50" h="50" backgroundimage="background.jpg" focusable="true"> ... </panel>
```

4.5.5 Paveikslukas

Mygtukas yra nefokusuojamas puslapio elementas, kuriam reikalinga labai nedaug atributų: pozicijos, dydžio ir nuorodos (7 lentelė). Mobilijame įrenginyje veikia tokių formatų paveikslukai: png, gif, bmp ir jpg.

7 lentelė. Paveiksluko elemento atributai ir jų aprašymas

Savybė	Paskirtis
source	Vaizduojamo paveiksluko nuoroda mobilijame įrenginyje.

24 pav. parodytas paveiksluko pavyzdys esantis lange.



Kraunasi...

24 pav. Paveiksluko elementas lange

Paveiksluko aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<image name="p1" x="5" y="5" source="pic1.png" />
```

4.5.6 Įvedimo laukelis

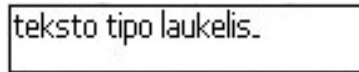
Įvedimo laukelis turi atributą nusakantį jo tipą. Galimi įvedimo laukelio tipai yra: tekstas, skaičiai, slaptažodis ir data. 8 lentelė apibrėžia galimus įvedimo laukelio atributus.

8 lentelė. Įvedimo laukelio elemento atributai ir jų aprašymas

Savybė	Paskirtis
disallow	Apibrėžia visus simbolius, kurių negalima įvesti į laukelį. Simboliai surašomi sekoje, pavyzdžiui galima uždrausti įvesti tokius simbolius kaip '#%?'
max	Didžiausia reikšmė, kurią vartotojas gali įvesti. Šis atributas galioja tik skaičių tipo laukeliui
maxchars	Maksimalus į teksto laukelį leidžiamų įvesti simbolių skaičius
maxday	Didžiausia diena (1-31), kurią vartotojas gali įvesti (galioja tiktai datos tipo laukeliui)
maxmonth	Didžiausias mėnesis (1-12), kurį vartotojas gali įvesti (galioja tiktai datos tipo laukeliui)
maxyear	Didžiausi metai, kuriuos vartotojas gali įvesti (galioja tiktai datos tipo laukeliui)
min	Mažiausia reikšmė, kurią vartotojas gali įvesti. Šis atributas galioja tik skaičių tipo laukeliui
minday	Mažiausia diena (1-31), kurią vartotojas gali įvesti (galioja tiktai datos tipo laukeliui)
minmonth	Mažiausias mėnesis (1-12), kurį vartotojas gali įvesti (galioja tiktai datos tipo laukeliui)
minyear	Mažiausi metai, kuriuos vartotojas gali įvesti (galioja tiktai datos tipo laukeliui)
day	Reikšmė nusakanti dieną, kuri rodoma datos laukelyje
month	Reikšmė nusakanti mėnesį, kuris rodomas datos laukelyje
year	Reikšmė nusakanti metus, kurie rodomi datos laukelyje
readonly	Jei ši reikšmė yra 'true', laukelio reikšmės redaguoti neleidžiama

type	Nusako įvedimo laukelio tipą. Galimos reikšmės yra: text password numbers date
value	Nusako pradinę laukelio reikšmę

25 pav. parodytas įvedimo laukelio pavyzdys, kuris yra teksto tipo.



25 pav. Elementas – teksto tipo įvedimo laukelis

Įvedimo laukelio aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<input w="140" h="40" x="40" y="220" maxchars="200"
type="text" value="teksto tipo laukelis" bordercolor=
"##000000" name="input1"/>
```

4.5.7 Tekstas

Tekstas yra nefokusuojamas lango elementas. Šis elementas gali atvaizduoti tekstą skirtingomis kryptimis, būtini atributai yra pozicija ir teksto reikšmė. 9 lentelė apibrėžia kitus teksto atributus.

9 lentelė. Teksto elemento atributai ir jų aprašymas

Savybė	Paskirtis
align	Apibrėžia teksto išlygiavimą, galimos reikšmės yra: left center right
direction	Teksto atvaizdavimo kryptis: galimos reikšmės: up down horizontal
text	Vaizduojama teksto reikšmė
wrap	Nusako ar tekstas turėtų būti perkeltas į naują eilutę, jei jis netelpa esančioje eilutėje
font	Vaizduojamo teksto šriftas
textcolor	Vaizduojamo teksto spalva

26 pav. Pavaizduotas dviejų teksto elementų pavyzdys, kurių pirmojo teksto stilius skiriasi nuo antrojo.



26 pav. Du teksto elementai esantys lange

Teksto elemento aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<label name="txt1" x="50" y="10" w="90" h="16"
text="Universitetas savo veiklos..." textcolor="#000000"
wrap="true" />
```

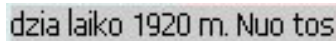
4.5.8 Judantis tekstas

Šis elementas yra nefokusuojamas ir atvaizduoja tekstą, judantį nurodyta kryptimi horizontaliai arba vertikalčiai. 10 lentelė apibrėžia judančio teksto atributus.

10 lentelė. Judančio teksto elemento atributai ir jų aprašymas

Savybė	Paskirtis
speed	Teksto judėjimo greitis, matuojamas pikselių skaičiumi per 0,1s
direction	Teksto judėjimo kryptis: galimos reikšmės: down up left right
text	Vaizduojama teksto reikšmė
font	Vaizduojamo teksto šriftas
textcolor	Vaizduojamo teksto spalva
source	Nuoroda į failą, iš kurio turi būti perskaitytas ir parodomas tekstas

27 pav. Pavaizduotas judančio teksto pavyzdys, kurio judėjimo kryptis yra į dešinę pusę.



27 pav. Elementas – judantis tekstas

Judančio teksto aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<scrolltext name="scrolltext1" x="50" bgcolor="#CCCCCC"
y="180" w="130" h="20" speed="1" direction="right"
```

```
text="Universitetas savo veiklos pradžia laiko 1920 m. Nuo tos
didzios datos iki..." />
```

4.5.9 Progreso juosta

Progreso juosta yra naudojama nusakyti tam tikro veiksmo būsenai. Galima progreso juostai priskirti tam tikrą tebevykstantį veiksmą, kad vizualiai būtų galima matyti jo progresą. 11 lentelė aprašo progreso juostos atributus.

11 lentelė. Progreso juostos elemento atributai ir jų aprašymas

Savybė	Paskirtis
action	Vykstantis veiksmas ir nurodantis elementui apie progresą
barcolor	Progreso juostos spalva
bgcolor	Juostos fono spalva
oncancel	Komanda, kuri turi būti vykdoma veiksmą atšaukus
oncomplete	Komanda, kuri turi būti vykdoma veiksmui pasibaigus
onerror	Komanda, kuri turi būti vykdoma įvykus klaidai
step	Skaičius, kuriuo progreso juostos būsenos reikšmė turėtų būti padidinta
value	Pradinė progreso juostos reikšmė. Didžiausia reikšmė (nurodanti, kad veiksmas baigėsi) yra 100

28 pav. parodyta progreso juosta, mėlyna spalva parodo jau įvykusio veiksmo dalį.



28 pav. Progreso juostos elemento pavyzdys

Progreso juostos aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<progress x="50" y="90" w="120" h="40" step="5"
action="sendmessage();" oncomplete="goback();" barcolor=
"#DE1410" />
```

4.5.10 Parinkčių mygtukai

Parinkčių mygtukai leidžia vartotojui tik vieną iš daugelio alternatyvų. 12 lentelė apibrėžia parinkčių mygtukų atributus.

12 lentelė. Parinkčių mygtukų elemento atributai ir jų aprašymas

Savybė	Paskirtis
upnavigation	Veiksmai vykdomu paėjimo į viršų metu
downnavigation	Veiksmai vykdomi paėjimo į apačią metu
disabled	Jeigu reikšmė yra ‚true‘, negalima keisti elemento būsenos ir elementas yra pašviesėjęs
buttonheight	Parinkčių mygtukų aukštis
buttonwidth	Parinkčių mygtukų plotis
initial	Pradžioje pažymėto mygtuko numeris, skaičiuoti pradedant nuo 0

Parinkčių mygtukų elementas viduje gali turėti bet kokių mygtukų kiekį, ir kiekvienas atskiras mygtukas gali turėti savo atributus (13 lentelė).

13 lentelė. Parinkčių mygtukų elemento vidinių mygtukų atributai ir jų aprašymas

Savybė	Paskirtis
text	Mygtuke vaizduojamas tekstas
setimage	Pažymėto mygtuko paveiksluko nuoroda
image	Mygtuko paveiksluko nuoroda
textcolor	Mygtuko teksto spalva
seltextcolor	Pažymėto mygtuko teksto spalva

29 pav. Parodytas parinkčių mygtukų pavyzdys, pirmasis yra pažymėtas, tai matosi vizualiai, nes jis išsiskiria ikonos ir teksto spalva.



29 pav. Parinkčių mygtukų elemento pavyzdys

Parinkčių mygtukų aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<radio name="radio1" x="10" y="90" w="230" h="50"
buttonheight="20" buttonwidth="230" initial="0">
```

```

        <radioitem      text="On"      setimage="rsbtn.bmp"      image=
"rbtn.bmp"      backgroundimage="1.bmp"      selectedbackimage="2.bmp"
textcolor="#ff0000"      seltextcolor="#00ff00" />
        <radioitem      text="Off"      setimage="rsbtn.bmp"      image=
"rbtn.bmp"      backgroundimage="1.bmp"      selectedbackimage="2.bmp"
textcolor="#ff0000"      seltextcolor="#00ff00" />
    </radio>

```

4.5.11 Funkcinis mygtukas

Funkcinis mygtukas yra nefokusuojamas ir nematomas lango elementas. Kadangi šis elementas yra nevaizduojamas ekrane, jam reikia nurodyti veiksmą, kuris bus atliekas jo aktyvavimo metu, o jis yra aktyvuojamas tam tikro mobilaus įrenginio klavišo paspaudimo metu (komandos). Standartiniai MIDP komandų tipus nusako visų įrenginio klavišų kodai (skaičiai), jų paspaudimo metu, atgalinis įrenginio ryšys bus nurodytos komandos įvykdymas. 14 lentelė aprašo funkcinio mygtuko atributus.

14 lentelė. Funkcinio mygtuko elemento atributai ir jų aprašymas

Savybė	Paskirtis
devicekeycode	Nusako mobilaus įrenginio klavišo kodą. Paspaudus atitinkamą klavišą, bus vykdomi veiksmai aprašyti atribute <code>onclick</code> . Vietoj skaičių, galimos šio atributo reikšmės yra: <code>back</code> <code>ok</code> <code>exit</code> <code>cancel</code> <code>help</code> <code>screen</code> <code>stop</code> <code>item</code> .
action	Komandos, kurios bus vykdomos mobilaus įrenginio klavišo paspaudimo metu

Funkcinio mygtuko aprašymo XML kalba pavyzdys pateikiamas žemiau:

```
<softkey keycode="165" onclick="goto('nextpage.xml');"/>
```

4.6 Vartotojo sąsajos funkcionalumo aprašymas

Siekiant vartotojo sąsajos aprašymui XML kalba suteikti funkcionalumą, apibrėšime XML žymenų atributuose galimas naudoti funkcijas (15 lentelė).

15 lentelė. XML dokumento galimų funkcijų aprašai

Funkcija	Aprašymas
<code>executetimeout(string script, number intervalMilli);</code>	Funkcija, kuri įvykdo <code>script</code> kodą po <code>intervalMilli32</code> intervalo (milisekundėmis) nuo jos iškvietimo.
<code>goto(string url);</code>	Ši funkcija atvaizduoja ekrane kito XML dokumento aprašytą vartotojo sąsają, dokumentas funkcijai paduodamas kaip parametras.
<code>exit();</code>	Išjungia programą.
<code>back();</code>	Pradedama rodyti prieš tai vaizduoto XML dokumento aprašomą sąsają.
<code>clickFocused();</code>	Ši funkcija įvykdo tuo metu pažymėto elemento veiksmus, kurie yra aprašyti elemento <code>onclick</code> atributu.
<code>showMessage(string text);</code>	Parodo naują langą su pranešimu, kuris funkcijai paduodamas per parametrus.
<code>readFile(string url);</code>	Skaito failą, į kuriį nuoroda paduodama funkcijai kaip parametras ir grąžina failo turinį.
<code>sendSMS(string phoneNumber, string message);</code>	Ši funkcija išsiunčia trumpąją žinutę vartotojui, kurio numeris yra <code>phoneNumber</code> , o pranešimo tekstas <code>message</code> .
<code>getDate(string format);</code>	Grąžina dabartinę datą, kurios formatas paduodamas funkcijai per parametrus (pavyzdžiui, yyyy:mm:dd)
<code>getTime(string format);</code>	Grąžina dabartinį laiką, kurio formatas paduodamas funkcijai per parametrus (pavyzdžiui, hh:mm:ss)
<code>getBatteryState();</code>	Ši funkcija grąžina skaičių indikuojantį kiek mobilaus įrenginio baterija yra pakrauta procentais.
<code>getSignalStrength();</code>	Grąžina mobilaus įrenginio signalo stiprumą procentais.
<code>playFile(string url, number count, number volume);</code>	Funkcija skirta vaizdo ir garso failų grojimui. Kaip parametrus funkcijai reikia paduoti nuorodą iki grojamo failo, skaičių kiek kartų kartoti grojimą ir skaičių nurodantį grojimo garsą procentais.
<code>sync(number showProgress);</code>	Šios funkcijos pagalba galima sinchronizuoti įrenginyje esančius failus su serveriu. Norint naudotis šia funkcija, jai reikalinga serverio pusė, kuri suprastų siunčiamas užklausas ir siųstų reikiamus failus. Ši funkcija siunčia serveriui turimų failų sąrašą, tada serveris sudaro sąrašą

	<p>failų, kurių nėra mobiliajame įrenginyje ir siunčia juos. Mobilusis įrenginys gavęs failus, perrašo egzistuojančius XML failus gautais. Parametras <code>showProgress</code> nusako ar failų siuntimo metu vartotojui turi būti rodoma progreso juosta (0 – nerodoma ir failai siunčiami fone, 1 – rodoma progreso juosta failų siuntimo metu).</p>
--	--

Be jau išvardintų funkcijų, egzistuoja `ELEMENT` raktažodis, kurio pagalba vartotojui suteikiama galimybė pakeisti bet kurio elemento visus atributus. Tarkim elemento vardas yra `elementas1`, ir norima pakeisti jo tekstą (atributas `text`), tuomet reikėtų rašyti tokią eilutę:

```
ELEMENT.elementas1.text = 'pakeistas tekstas';
```

O šią eilutę parašius pavyzdžiui mygtuko `onclick` attribute, elemento `elementas1` tekstas pasikeistų paspaudus mygtuką.

Keleto XML failų, aprašančių vartotojo sąsają, turinys ypa pateiktas 10C priede.

4.7 Programos veiksmų seka

Veiksmai reikalingi tam, kad būtų galima teisingai pavaizduoti XML failuose aprašytą vartotojo sąsają:

1. Visas XML failo turinys perskaitomas ir analizuojamas, elementai ir jų atributai saugomi operatyvioje atmintyje.
2. Sukuriamas kiekvienas elementas, jam priskiriant atributų reikšmes, tai gali būti vaizduojamas tekstas, nuoroda į paveiksluką ir kt.
3. Grafiniai elementai piešiami ekrane, pradedant nuo visus gaubiančio elemento – lango.
4. Klausomasi funkcinų mygtukų įvykių, kurie apibrėžia navigaciją į sekantį ar prieš tai rodytą langą, ar kitas funkcijas.

Tas pačias funkcijas atliekančių skirtingų įrenginių mygtukų, bet turinčių skirtingus klavišų kodus problema programinėje įrangoje sprendžiama iš statinės lentelės pagal įrenginio tipą paimant tam tikrą funkciją atliekančio klavišo kodą. Tuomet vartotojas rašydamas XML, vietoje klavišo kodo turi paduoti funkciją, kuria atlieka klavišas, pavyzdžiui:

```
<softkey keycode="up" onclick="goto('nextpage.xml');"/>
```

16 lentelė apima visas galimas funkcijas. Tačiau jei tam tikra reikalinga funkcija nėra aprašyta, galima paduoti klavišo kodą.

16 lentelė. Raktažodžiai atitinkantys mygtukus pagal atliekamą funkciją

Savybė	Paskirtis
ok	Veiksmas vykdomas paspaudus patvirtinimo mygtuką
cancel	Atitinkamas veiksmas bus įvykdytas paspaudus atšaukimo mygtuką
exit	Išjungimo mygtukas
back	Navigacija į ankstesnįjį langą
help	Pagalba
up	Rodyklės į viršų mygtukas
down	Rodyklės į apačią mygtukas
left	Rodyklės į kairę mygtukas
right	Rodyklės į dešinę mygtukas

4.8 Sistemos privalumai

4.8.1 Galimybė kurti nestandartinius elementus

Nestandartinius elementus galima kurti XML dokumente įvairiai išdėstant skirtingus elementus (paveikslukus, tekstą ir kt.) ir naudojant vartotojui iš XML prieinamas funkcijas. 30 pav. pateikti elementai vartotojui paspaudus mygtukus į kairę arba dešinę, vaizdas persislenka ir parodomas kitas paveikslukas. Elementai yra sudaryti iš paveikslukų, tekstų ir mygtukų.



30 pav. Nestandartiniai vartotojo sąsajos elementai

Tokia sistema teikia paprastą, intuityvią programos turinio paslaugų navigaciją bei priėjimą prie pagrindinių įrenginio funkcijų. Pavyzdžiui, vartotojas gali perklausyti garsus, peržiūrėti vaizdinę medžiagą, skaityti jos aprašymą, peržiūrėti paveikslukus su aprašymu vien tik keleto klavišų paspaudimu.

Technologija susideda iš mažos programos mobiliajame įrenginyje. Įrenginyje programa gali būti įdiegta naudojantis Bluetooth, Infra-red technologijomis, ar atsiųsta per internetą. Dinaminė vartotojo sąsaja (visas vartotojui matomas programos turinys ir funkcionalumas) XML ir kitų reikalingų failų pavidalu gali būti atnaujinti naudojantis internetu ir iš anksto serveryje padėtu programos turiniu.

Kiti privalumai yra tai, kad vartotojui lengva naudotis sistema, loginiai paprasti meniu ir navigacija, reikalaujanti tik keleto paspaudimų, programos išvaizda ir funkcionalumas skirtinguose įrenginiuose išlieka tokie patys, greita navigacija tarp langų, veikia daugelyje įrenginių ir platformų, XML ir kiti pagalbiniai failai saugomi lokaliai, nereikia nuolatinio interneto ryšio, kurti XML failus nereikia daug laiko, XML scenarijai yra lengvai suprantami, lengva modifikuoti ir palaikyti.

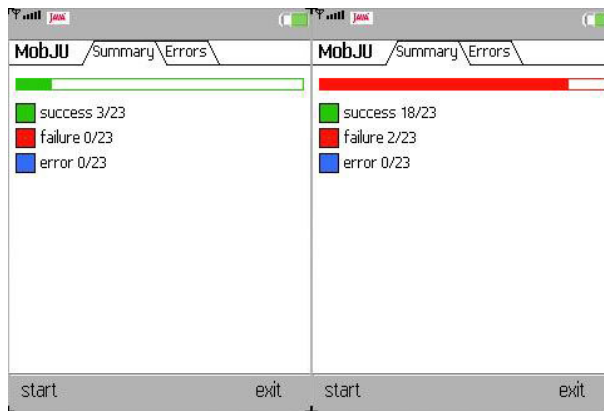
4.9 Testavimo galimybių apžvalga

4.9.1 J2ME vienetų testavimas

J2ME taikomosios programos automatizuotam vienetų testavimui galima panaudoti Sony Ericsson Mobile JUnit įrankį.

Programa yra suskaidyta į klases, todėl kiekvienai klasei ar klasių grupei bus rašomi atskiri testavimo atvejai. Kiekvienas sistemos paketas turi tam tikrą testinių atvejų skaičių ir įeina į testų rinkinį. Vienetų testų sukūrimas sumažins laiko, reikalingo kodo nagrinėjimui, sąnaudas, leis saugiai keisti tam tikras kodo vietas, užtikrinant, kad kitos sistemos dalys veikia be klaidų, bet kada pakartotinai įvykdyti testus su minimaliomis pastangomis.

Paveldėdamas testuojamos programos pagrindinę klasę, Mobile JUnit suteikia priėjimą prie visų klasės metodų, taip suteikdamas galimybę efektyviai ištestuoti programinę įrangą. Mobile JUnit turi vartotojo sąsają, kurioje rodomas vykdomo testo numeris, bei statistika: kiek testų jau įvykdyta, kiek įvyko be klaidų ir kitą informaciją. Emuliatoriaus vaizdas su Mobile JUnit testų vykdymo metu pateiktas 31 pav.



31 pav. Sony Ericsson Mobile JUnit įrankio pagalba vykdomi testai

4.9.2 Vartotojo sąsajos testavimas

Grafinės vartotojo sąsajos testavimas yra sudėtingas procesas ir dažniausiai šį procesą galima automatizuoti tik dalinai. Šiame skyriuje pateikiama keletas dinaminės vartotojo sąsajos testavimo galimybių.

XML teikiamo funkcionalumo pagalba panaudojant išskirtines ir tik testavimui skirtas funkcijas `upnavigation()` ir `downnavigation()`, kurios sufokusuoja aukščiau arba žemiau esantį elementą atitinkamai. Tokiu būdu pereinant per visus lango elementus iš eilės, vykdyti kitą funkciją - `executefocused()`, tokiu būdu iškviečiant veiksmus, kurie yra aprašyti pažymėto elemento atribute `onclick`. Tai gali būti navigacijos ar bet koks kitas veiksmas. Testavimo tikslais galima sukurti atskirus XML failus, kurie vykdytų juose aprašytas funkcijas ir taip testuotų tam tikras vartotojo sąsajos dalis.

Pavyzdžiui, testuojant įvedimo laukelio elementą ir užėjus ant jo, XML funkcijų pagalba priskirti tam tikrą reikšmę, pavyzdžiui jei įvedimo laukelis yra skaičių tipo, jam priskirti tekstą ir po tokio priskyrimo patikrinti ar laukelio reikšmė savyje neturi simbolių, kurie nėra skaičiai. Toks testas leistų patikrinti įvedimo laukelio tipo teisingumą ir užtikrinti, kad neleidžiama įvesti draudžiamų simbolių.

Grafinę vartotojo sąsają, aprašytą XML kalba, gan nesudėtinga ir efektyvu būtų testuoti vykdant tokią veiksmų seką:

- ✓ Pirmą kartą testuotojas patikrina peržiūrėdamas visus langus akimis ir jei visi elementai yra išdėstyti teisingai ir vartotojo sąsaja neturi jokių klaidų, tuomet `upnavigation()`, `downnavigation()` ir `executefocused()` funkcijų pagalba einama per visus langus ir kiekvienas langas yra

nufotografuojamas visose būsenose. Langas yra skirtingoje būsenoje, kai skirtingas jo elementas yra pažymėtas.

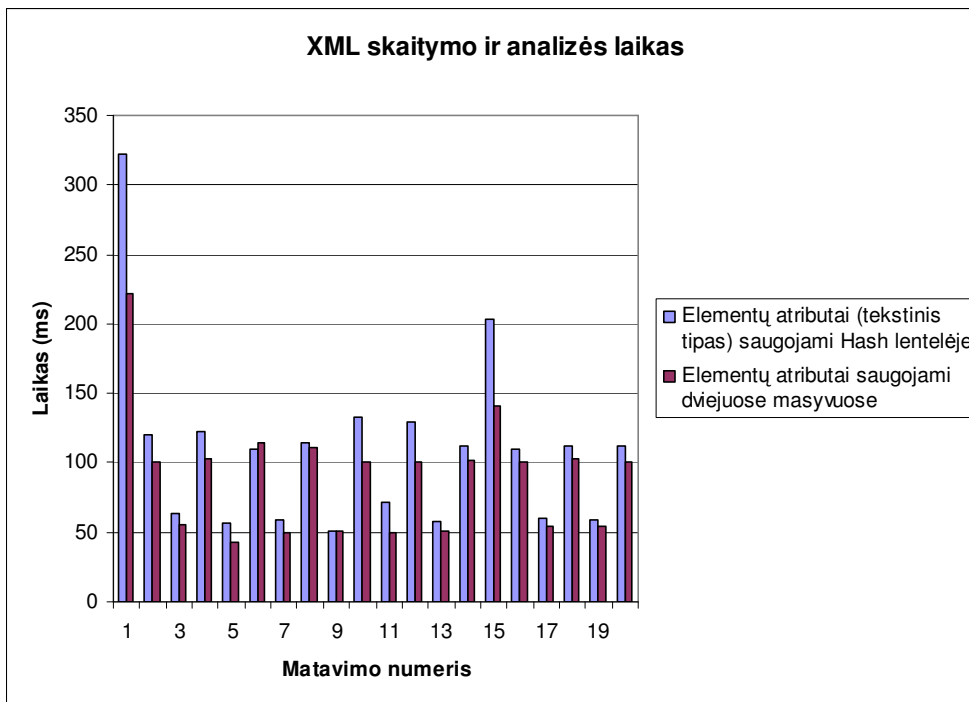
- ✓ Nufotografuotas Canvas objekto vaizdas (viskas, kas matoma lange) paverčiamas į baitų masyvą ir išsaugomas, šie duomenys laikomi šabloniniais.
- ✓ Kitą kartą vykdant testus, atliekama ta pati procedūra, t.y. einama per visų langų visas būsenas, duomenys verčiami į baitų masyvus ir tuomet lyginami su šabloniniais duomenimis (tais duomenimis, kurie buvo patikrinti vartotojo patį pirmą kartą). Atitinkamiems baitų masyvams nesutapus, laikoma, kad įvyko vartotojo sąsajos klaida.

5 EKSPERIMENTINĖ DALIS

Šiame skyriuje pateikiama sukurtos ir įdiegtos programinės įrangos bei jos patobulinimų eksperimentinis tyrimas.

5.1 XML analizatoriaus veikimo greičio tyrimas ir galimi patobulinimai

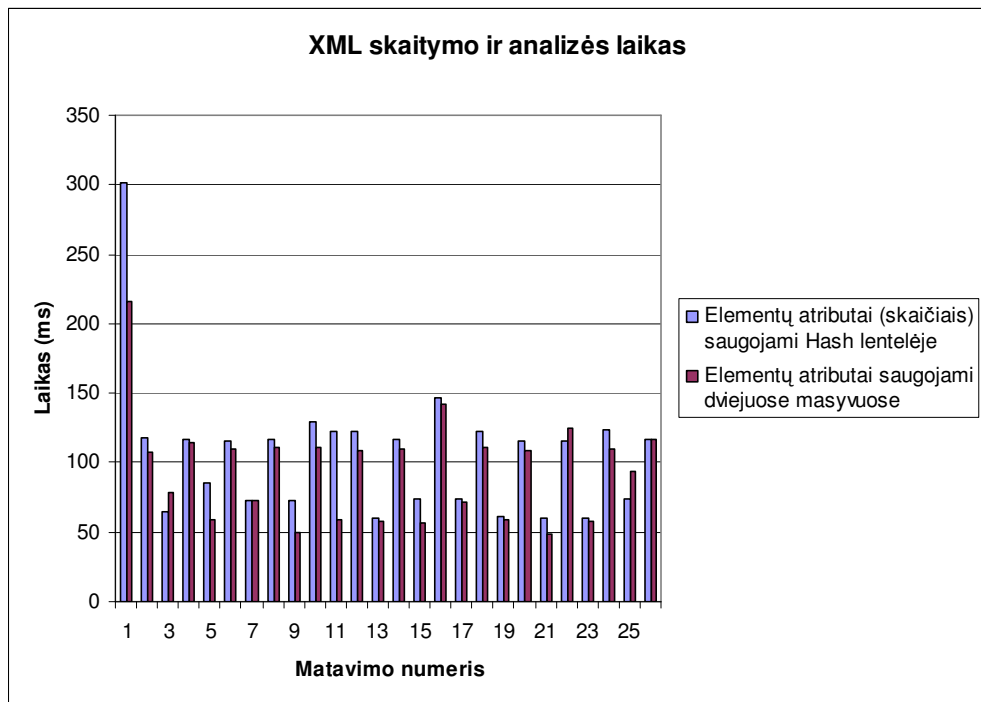
Mobiliųjų renginių resursai yra riboti, todėl yra svarbu sutaupyti kiekvieną milisekundę procesoriaus darbo laiko. XML analizavimas užima palyginti nemažai laiko, todėl šiame skyriuje yra lyginama keletas skirtingų XML analizės būdų, naudojant skirtingus objektus. 32 pav. lyginama koks yra XML analizatoriaus veikimo laiko skirtumas naudojant J2ME objektą hash lentelę XML elementų atributų reikšmių saugojimui (tekstiniu formatu) ir paprastus du masyvus.



32 pav. XML failo skaitymo ir analizės laikų palyginimas saugant elementų atributus skirtingais būdais

Iš palyginimo galima daryti išvadą, kad naudojant masyvus galime sutaupyti iki 50-100 milisekundžių. Toks laiko tarpas yra gan didelis turint omenyje, kad tiek galima sutaupyti kiekvieno perėjimo iš vieno lango į kitą metu. Kitas XML skaitymo ir analizės laiko

matavimo eksperimentas buvo darytas su dviem masyvais ir taip pat hash lentele, tačiau joje atributus saugant jau ne tekstiniu formatu, bet perskaičius ir pavertus juos į skaičius (33 pav.).



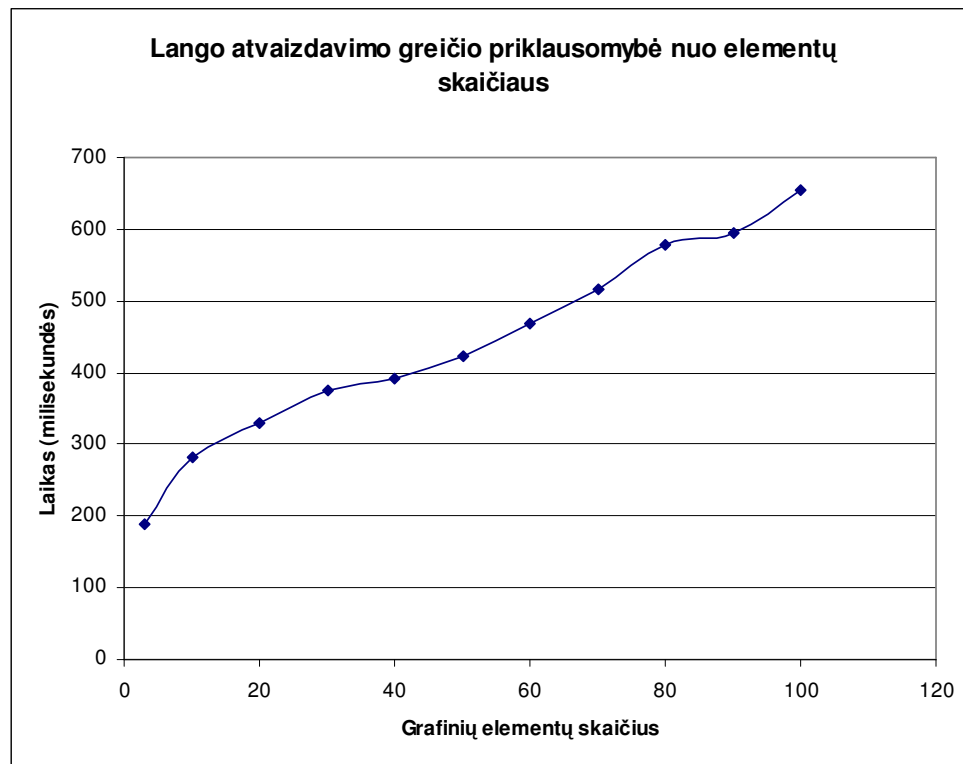
33 pav. XML failo skaitymo ir analizės laikų palyginimas saugant elementų atributus skirtingais būdais

XML skaitymas ir analizė naudojant hash lentelės objektą ir šio eksperimento metu pasirodė lėtesnis nei tos pačios operacijos atlikimas naudojant du masyvus, tačiau šiuo būdu įgyvendintas skaitymas buvo šiek tiek greitesnis, nei lentelėje saugant tekstinio tipo atributus.

Taigi, atlikus tokius bandymus, galima daryti išvadą, kad efektyviausias elementų atributų saugojimo ir išrinkimo būdas yra tuomet, kai atributai saugomi masyvuose.

5.2 Dinaminės grafinės vartotojo sąsajos elementų piešimo tyrimas

Šiame skyriuje bus atliekami eksperimentai ir palyginimas piešimo laikų, kai lange piešiami keli ir kelios dešimtys elementų (34 pav.). Tokiu būdu tikrinamas piešimo greičio priklausomybė nuo vaizduojamų elementų kiekio.



34 pav. Lango parodymo ekrane greičio priklausomybė nuo elementų kiekio

Iš grafiko galima pastebėti, kad elementų atvaizdavimo greitis ekrane maždaug tolygiai lėtėja didėjant atvaizduojamų elementų kiekiui. Vaizduojant virš 100 elementų ekrane, jų parodymo laikas artėja prie 1 sekundės, o tai yra gan ilgas laiko tarpas vartotojo navigacijai tarp skirtingų langų. Optimaliausias grafinių elementų atvaizdavimo laikas būtų tuomet, kai ekrane piešiama iki 10 elementų.

5.3 Testavimo galimybių tyrimas

Automatizuotas testavimas turėtų užtikrinti aukštesnį testavimo efektyvumą, t. y. automatizuoto testavimo pagalba turėtų būti galima testuoti visiems įrenginiams ir visoms programoms būdingas savybes, tokias kaip perėjimas iš vieno lango į kitą, elementų atvaizdavimas, jų vieta lange ir kt.

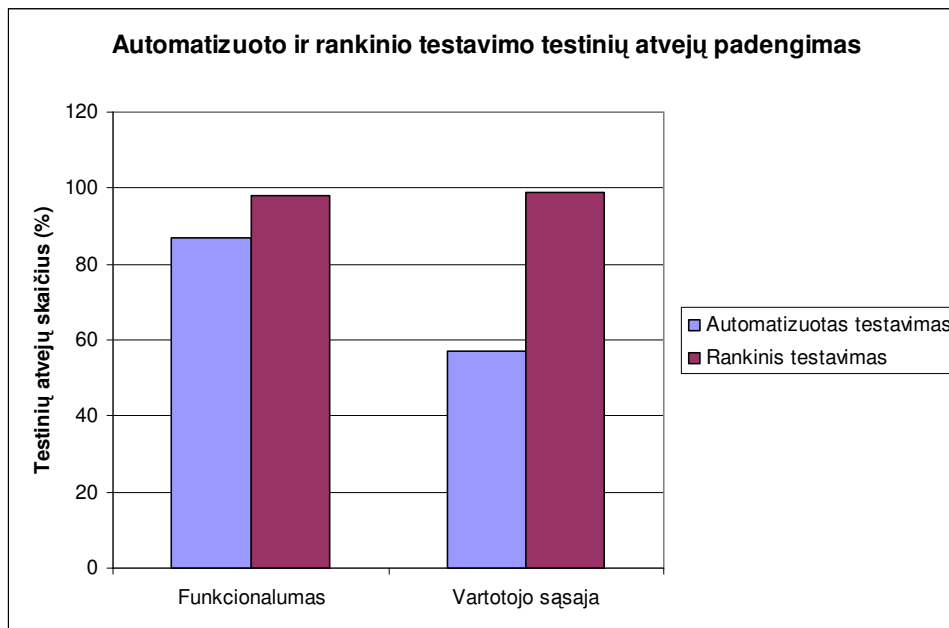
Grafinės vartotojo sąsajos testavimo pilnai automatizuoti beveik neįmanoma, darbe vartotojo sąsaja buvo testuota panaudojant XML teikiamomis funkcijomis `upnavigation()`, `downnavigation()` ir `executefocused()` tokiu būdu išskviečiant veiksmus, kurie yra aprašyti pažymėtų elementų `onclick` atributuose. Dažniausiai tai – navigacijos į kitą langą veiksmas. Testavimo tikslais buvo sukurtas atskiras

XML failas, kuris skaitomas vienos iš testavimo klasės ir faile aprašytos funkcijos buvo vykdomos Mobile JUnit įrankio pagalba. Taigi, tokiu atveju kiekvienam vartotojo sąsajos aprašymui XML kalba, reikalingas kitas failas, aprašantis testavimo instrukcijas.

Rankinis testavimas vis tiek yra būtinas norint įsitikinti, kad programa mobiliajame įrenginyje veikia teisingai. Ypatingai testavimo atvejų rašymas ir jų rankinis vykdymas naudingas siekiant ištestuoti atskirų įrenginių ypatybes.

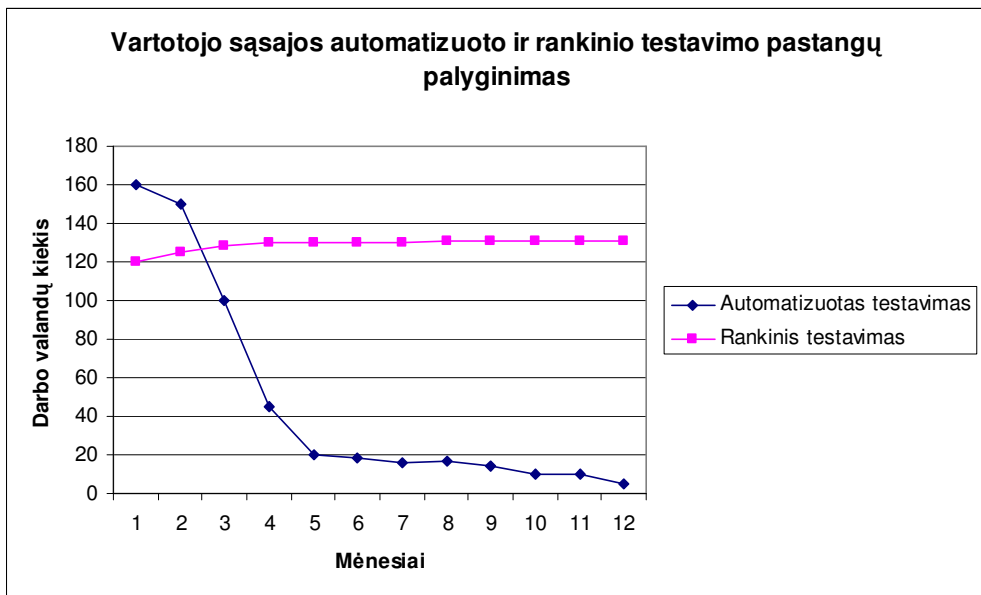
Pavyzdžiui, testuojant įvedimo laukelio elementą XML funkcijų pagalba, jam buvo priskirtos tam tikros reikšmės ir tokiu būdu tikrinama ar vartotojas negali įvesti neleidžiamų reikšmių ir pan. Tokie testai leistų patikrinti įvedimo laukelio tipo teisingumą ir užtikrinti, kad neleidžiama įvesti draudžiamų simbolių, taip galima patikrinti laukelio poziciją, tačiau norint patikrinti ar visi elementai ekrane rodomi teisingai, jų spalvos, rėmeliai ir kt., rankinis testavimas yra reikalingas.

Kaip galima matyti iš paveikslo (35 pav.), funkcionalumą testavimą automatizuoti yra lengviau, nei pačios vartotojo sąsajos testavimą. Norint pilnai ištestuoti šiame darbe sukurtą programinę įrangą, rankinis testavimas taip pat yra būtinas. Funkcinį testavimą taip pat apima J2ME programos vienetų testavimas įrankio Mobile JUnit pagalba.



35 pav. Automatizuoto ir rankinio testavimo testinių atvejų padengimas

Grafinę vartotojo sąsają, aprašytą XML kalba, gan nesudėtinga testuoti pirmą kartą peržiūrint ir įsitikinant viskas veikia taip, kaip ir turėtų, tuomet įsimenant kiekvieno lango statinį vaizdą ir sekančius kartus, padarius tam tikrus pakeitimus, tikrinti ar grafinės sąsaja nepakito (kaip aprašyta 4.9.2 skyriuje). Tuomet kiekvienas pakartotinis grafinės vartotojo sąsajos testavimas reikalauja tik minimalių pastangų (36 pav.).



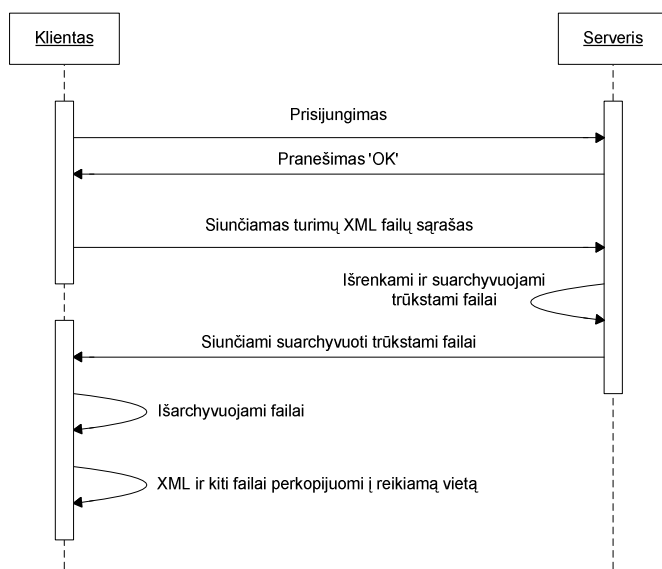
36 pav. Vartotojo sąsajos automatizuoto ir rankinio testavimo palyginimas

Matome, kad siekiant užtikrinti ilgalaikį programinės įrangos palaikymą, žymiai efektyviau yra sukurti automatinius testus ir vėliau juos naudoti, vietoje nuolatos didelių pastangų reikalaujančio rankinio testavimo.

6 ATEITIES PATOBULINIMAI

6.1 Sinchronizacija su serveriu

Vienas iš rimtų sistemos patobulinimų gali būti kliento-serverio ryšys, kurio pagalba mobilusis įrenginys galėtų gauti naujus XML, paveikslukų ir kitus failus iš serverio, J2ME taikomoji programa fone juos perkopijuotų į direktoriją, kurioje yra kiti XML failai ir tokiu būdu vartotojas galėtų visiškai atnaujinti grafinę sąsają bei jos funkcionalumą.



37 pav. Kliento ir serverio PĮ galimos komunikacijos diagrama

37 pav. pateiktas galima kliento-serverio komunikacija. Vartotojui nauja sąsaja galėtų būti siunčiama nustatytais laiko intervalais, arba vartotojui pačiam paspaudus atnaujinimo mygtuką.

6.2 Kiti patobulinimai

Kiti galimi tokios architektūros patobulinai:

- ✓ Funkcionalumo praplėtimas – naujų funkcijų įgyvendinimas, kurios būtų prieinamos iš XML failo, pavyzdžiui funkcijos operacijoms su tekstiniais duomenimis, matematinės funkcijos, informacija apie atmintį, žinučių kiekį telefone ir kt.
- ✓ Papildomų naujų elementų palaikymas J2ME programoje
- ✓ Universalus testavimo karkaso dinaminei vartotojo sąsajai, paremtai XML kalba, sukūrimas

7 IŠVADOS

Darbe buvo sukurta mobiliųjų įrenginių dinaminė vartotojo sąsaja ir, remiantis aprašyta technologija, pasiūlytas šios sąsajos automatizuotas testavimo būdas. Aprašytas grafinės vartotojo sąsajos kūrimo būdas teikia paprastą, intuityvią programos turinio paslaugų navigaciją bei priėjimą prie pagrindinių mobilaus įrenginio funkcijų. Technologija sudaryta iš mažos J2ME programos mobiliajame įrenginyje, kuri skaito ir atvaizduoja XML aprašytą sąsają. Dinaminė vartotojo sąsaja (visas vartotojui matomas programos turinys ir funkcionalumas) XML ir kitų reikalingų failų pavidalu gali būti greitai ir lengvai pakeista ar atnaujinta. Darbe tiriamas elementų piešimo, XML skaitymo, navigacijos tarp skirtingų langų greičiai. Gauti rezultatai parodė, kad sąsajos atvaizdavimo laiką galima sutrumpinti optimizuojant XML failų skaitymą, bei ekrane piešiant ne daugiau 20 elementų.

Sistemos pagrindiniai teikiami privalumai:

- ✓ Vartotojui lengva naudotis, loginiai paprasti meniu ir greita navigacija, reikalaujanti tik keleto paspaudimų
- ✓ Veikia daugelyje įrenginių ir platformų, tokia pati programos išvaizda ir funkcionalumas skirtinguose įrenginiuose
- ✓ Dinaminio turinio greitas pateikimas, valdymas, XML ir kiti pagalbinių failai saugomi lokaliai, nereikia nuolatinio interneto ryšio
- ✓ XML failų kūrimas užtrunka sąlyginai mažai laiko, XML turinys yra lengvai suprantamas, lengva modifikuoti ir palaikyti

Grafinės vartotojo sąsajos testavimas yra gan sudėtingas procesas ir kartais net neįmanomas. Remiantis darbe aprašytu vartotojo sąsajos kūrimo būdu, buvo pateikti sąsajos testavimo automatizavimo būdai. Vartotojo sąsajai, aprašytai XML kalba, testuoti buvo pasiūlytas sprendimas pirmą kartą rankiniu būdu peržiūrėti ir įsitikinti, kad viskas veikia taip, kaip ir turėtų, tuomet įsiminti kiekvieno lango statinį vaizdą ir sekančių testavimų metu, padarius tam tikrus pakeitimus, tikrinti ar vartotojo sąsaja nepakito. Tuomet kiekvienas pakartotinis grafinės vartotojo sąsajos testavimas reikalauja tik minimalių pastangų.

Sukurta sistema gali būti patobulinta įdiegiant tokias technologijas:

- ✓ Kliento-serverio architektūra XML ir kito reikalingo turinio atnaujinimui
- ✓ Naujų elementų palaikymas J2ME programoje
- ✓ Naujų funkcijų palaikymo įgyvendinimas
- ✓ Universalus vartotojo sąsajos testavimo karkaso, paremto XML kalba, sukūrimas

8 LITERATŪRA

- [1]. Mobile phone features, Wikipedia [interaktyvus]. 2007, balandis [žiūrėta 2007-04-05]. Prieiga per internetą: http://en.wikipedia.org/wiki/Mobile_phone_features
- [2]. Canvas Keycodes [interaktyvus]. 2007, balandis [žiūrėta 2007-04-05]. Prieiga per internetą: http://www.j2meforums.com/wiki/index.php/Canvas_Keycodes
- [3]. The Joy of XUL, Mozilla developer center [interaktyvus]. 2007, balandis [žiūrėta 2007-04-12]. Prieiga per internetą: http://developer.mozilla.org/en/docs/The_Joy_of_XUL
- [4]. XUL Questions and Examples, XUL Planet [interaktyvus]. 2006 [žiūrėta 2007-04-12]. Prieiga per internetą: <http://www.xulplanet.com/tutorials/xulqa/>
- [5]. Bambookit, Bambookit GUI [interaktyvus]. 2006 žiūrėta 2007-04-12]. Prieiga per internetą: <http://www.bambookit.com/>
- [6]. Thinlet [interaktyvus]. 2002-2007 [žiūrėta 2007-04-12]. Prieiga per internetą: <http://www.thinlet.com/>
- [7]. Mobile platform statistics [interaktyvus]. 2006, birželis [žiūrėta 2007-04-12]. Prieiga per internetą: <http://alindh.iki.fi/2006/06/27/mobile-platform-statistics/>
- [8]. Graphical User Interface, TechTarget [interaktyvus]. 2006, spalio [žiūrėta 2007-04-12]. Prieiga per internetą: http://searchvb.techtarget.com/sDefinition/0,,sid8_gci213989,00.html
- [9]. J2ME Tech Tips, Sun Developer Network [interaktyvus]. 2001, lapkritis [žiūrėta 2007-04-12]. Prieiga per internetą: <http://java.sun.com/developer/J2METechTips/2001/tt1114.html>
- [10]. M. Kroll and S. Haustein, J2ME Application Development, Chapter 3: MIDP Programming. Sams Publishing [žiūrėta 2007-04-20]. ISBN:0-672-323095-9.
- [11]. Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation [interaktyvus]. 2004, vasaris [žiūrėta 2007-04-20]. Prieiga per internetą: <http://www.w3.org/TR/REC-xml/>
- [12]. XML, Wikipedia [interaktyvus]. 2005 [žiūrėta 2007-04-20]. Prieiga per internetą: <http://en.wikipedia.org/wiki/XML>
- [13]. J. Knudsen, Parsing XML in J2ME [interaktyvus]. 2002, kovas [žiūrėta 2007-04-20]. Prieiga per internetą: <http://developers.sun.com/techttopics/mobility/midp/articles/parsingxml/>

- [14]. T. Kunneth, Mobile Memories: The MIDP Record Management System [interaktyvus]. 2004 lapkritis [žiūrēta 2007-04-20]. Prieiga per internetā: <http://today.java.net/pub/a/today/2004/11/16/J2ME-3.html>
- [15]. E. Giguere, Databases and MIDP, Part 1: Understanding the Record Management System [interaktyvus]. 2004 vasaris [žiūrēta 2007-04-20]. Prieiga per internetā: <http://developers.sun.com/techttopics/mobility/midp/articles/databaserms/>
- [16]. Open Mobile Alliance, Building an Industry-Wide Mobile Data Synchronization Protocol [interaktyvus]. 2001 [žiūrēta 2007-04-20]. Prieiga per internetā: <http://www.openmobilealliance.org/tech/affiliates/syncml/whitepaper.pdf>
- [17]. D. Heintzman, SyncML [interaktyvus]. 2002 saisis [žiūrēta 2007-04-25]. Prieiga per internetā: http://www.openmobilealliance.org/tech/affiliates/syncml/introducing_syncml_29jan02_douglas_heintzman.pdf
- [18]. Q. H. Mahmoud, Getting Started with Data Synchronization Using SyncML [interaktyvus]. 2004 rugsējis [žiūrēta 2007-04-25]. Prieiga per internetā: <http://developers.sun.com/techttopics/mobility/midp/articles/syncml/>
- [19]. D. Buchmann, SyncML and its java implementation sync4j [interaktyvus]. 2002 rugsējis [žiūrēta 2007-04-25]. Prieiga per internetā: <http://sync4j.funambol.com/RandD/David%20Buchmann.pdf>
- [20]. GUI software testing, Wikipedia [interaktyvus]. 2007 [žiūrēta 2007-04-25]. Prieiga per internetā: http://en.wikipedia.org/wiki/GUI_software_testing
- [21]. P. Gerrard, Testing GUI Applications [interaktyvus]. 1997, lapkritis [žiūrēta 2007-04-25]. Prieiga per internetā: <http://www.gerrardconsulting.com/GUI/TestGui.html>
- [22]. Mobile development, Wikipedia [interaktyvus]. 2007 [žiūrēta 2007-04-25]. Prieiga per internetā: http://en.wikipedia.org/wiki/Mobile_development
- [23]. B. Marick and P. Atria, The Test Manager at the Project Status Meeting [interaktyvus]. 1997 [žiūrēta 2007-04-28]. Prieiga per internetā: <http://www.testing.com/writings/status/status.html>
- [24]. A. M. Memon, A COMPREHENSIVE FRAMEWORK FOR TESTING GRAPHICAL USER INTERFACES. Submitted to the Graduate Faculty of Arts and Sciences in partial fulfillment of the requirements for the degree of Doctor of Philosophy, University of Pittsburgh, 2001.
- [25]. J2MEUnit, A J2ME Unit Testing Framework [interaktyvus]. 2006 [žiūrēta 2007-04-28]. Prieiga per internetā: <http://j2meunit.sourceforge.net/doc.html>

- [26]. J. Pleumann, Antenna - An Ant-to-End Solution For Wireless Java [interaktyvus]. 2002-2006 [žiūrėta 2007-04-28]. Prieiga per internetą: <http://antenna.sourceforge.net/>
- [27]. Sony Ericsson Mobile JUnit 1.0 for Java ME CLDC phones [interaktyvus]. 2006, liepa [žiūrėta 2007-04-28]. Prieiga per internetą: http://developer.sonyericsson.com/site/global/newsandevents/latestnews/newsjuly06/p_mobile_juint1.0_javame_cldc.jsp

9 TERMINŲ IR SANTRUMPŲ ŽODYNAS

J2ME	- Java Micro Edition
J2SE	- Java Standard Edition
J2EE	- Java Enterprise Edition
MIDP	- Mobile Information Device Profile
JVM	- Java Virtual Machine
PDA	- Personal Digital Assistant
JVM	- Java Virtual Machine
XML	- eXtensible Markup Language
XUL	- XML User Interface Language
XBL	- eXtensible Bindings Language
HTML	- Hypertext Markup Language
DHTML	- Dynamic Hypertext Markup Language
CSS	- Cascading Style Sheets
CLDC	- Connected Limited Device Configuration
API	- Application Programming Interface
GUI	- Graphical User Interface
PP	- Personal Profile

10 PRIEDAI

A XUL vartotojo sąsajos aprašai

Išsiskleidžiančių laukelių aprašas XUL kalboje:

```
<vbox datasources="animals.rdf" ref="http://www.some-fictitious-zoo.com/all-animals">
  <template>
    <rule iscontainer="true">
      <hbox align="left" uri="rdf:*">
        <label value="rdf:http://www.some-fictitious-zoo.com/rdf#name"/>
      </hbox>
    </rule>
    <rule>
      <conditions>
        <content uri="?uri"/>
        <member container="?uri" child="?animal"/>
        <triple subject="?animal"
          predicate="http://www.some-fictitious-zoo.com/rdf#name"
          object="?name"/>
      </conditions>
      <action>
        <menulist>
          <menupopup>
            <menuitem uri="?animal" label="?name"/>
          </menupopup>
        </menulist>
      </action>
    </rule>
  </template>
</vbox>
```

Medžio elemento aprašas XUL kalboje:

```
<tree rows="10" datasources="animals.rdf"
      ref="http://www.some-fictitious-zoo.com/all-animals">
  <treecols>
    <treecol id="name" label="Name" primary="true" flex="1"/>
    <treecol id="favorite" label="Favorite" flex="1"/>
  </treecols>
  <template>
    <rule>
      <conditions>
        <content uri="?uri"/>
        <member container="?uri" child="?animal"/>
        <triple subject="?human"
          predicate="http://www.some-fictitious-zoo.com/rdf#favoriteAnimal"
          object="?animal"/>
        <triple subject="?animal"
```



```

        predicate="http://www.some-fictitious-zoo.com/rdf#name"
        object="?name"/>
    <triple subject="?human"
        predicate="http://www.some-fictitious-zoo.com/rdf#name"
        object="?humanname"/>
</conditions>
<action>
    <treechildren flex="1">
        <treeitem uri="?animal">
            <treerow>
                <treecell label="?name"/>
                <treecell label="?humanname"/>
            </treerow>
        </treeitem>
    </treechildren>
</action>
</rule>
<rule>
    <conditions>
        <content uri="?uri"/>
        <member container="?uri" child="?animal"/>
        <triple subject="?animal"
            predicate="http://www.some-fictitious-zoo.com/rdf#name"
            object="?name"/>
    </conditions>
    <action>
        <treechildren flex="1">
            <treeitem uri="?animal">
                <treerow>
                    <treecell label="?name"/>
                    <treecell label=""/>
                </treerow>
            </treeitem>
        </treechildren>
    </action>
</rule>
</template>
</tree>

```

B SyncML protokolo pranešimo pavyzdys

Kliento ir serverio sinchronizacijos pranešimų pavyzdys (17 lentelė).

17 lentelė. Kliento ir serverio sinchronizacijos pranešimai

Kliento sync pranešimas	Serverio sync pranešimas
<pre> <SyncML> <SyncHdr> <VerDTD>1.1</VerDTD> </pre>	<pre> <Sync> <CmdID>5</CmdID> <Target><LocURI>dev_db/</LocURI></Target> </pre>

<pre> <VerProto>SyncML/1.1</VerProto> <SessionID>1029143392160</SessionID> <MsgID>1</MsgID> <Target><LocURI>http://localhost/sync</LocURI> </Target> <Source><LocURI>IMEI:123445</LocURI> </Source> <Cred> <Meta> <Type xmlns='syncml:metinf'>syncml:auth-basic </Type> <Format>b64</Format> </Meta> <Data>dXNlcjpwYXNz</Data> </Cred> </SyncHdr> <SyncBody> <Alert> <CmdID>1</CmdID> <Data>200</Data> </Item> <Target><LocURI>server_db</LocURI> </Target> <Source><LocURI>dev_db</LocURI></Source> <Meta><Anchor xmlns='syncml:metinf'> <Last>1004589132042</Last> <Next>2029143393370</Next></Anchor> </Meta> </Item> </Alert> <Final/> </SyncBody> </SyncML> </pre>	<pre> <Source><LocURI>server_db</LocURI></Source> <Replace> <CmdID>6</CmdID> <Item> <Target><LocURI>c1</LocURI></Target> <Source><LocURI>1</LocURI></Source> <Meta> <Type xmlns='syncml:devinf'>text/plain</Type> </Meta> <Data>arbitrary data that replaces client item c1 </Data> </Item> </Replace> <Delete> <CmdID>6</CmdID> <Item> <Target><LocURI>c3</LocURI></Target> <Source><LocURI>23ab3</LocURI></Source> <!--reikalingas tik ID įrašui ištrinti --> </Item> </Delete> </Sync> </pre>
---	---

C Vartotojo sąsajos aprašas XML kalba

Failas Main.xml:

```

<window backgroundimage="/bg_new.png">
  <image name="header" source="/header.png" x="0" y="7" focusable="false" />
  <list name="top" h="80" w="176" x="30" y="80" drawborder="false">

```

```

        <panel name="topwall" x="0" y="0" w="176" h="80" onclick="goto('/fun/w/g/1.xml');"
drawborder="false" backgroundimage="/teaserback_01.png"
selectedbackgroundimage="/teaserback_02.png" >
        <image source="/fun/w/g/1p.png" x="5" y="5" />
        <label text="Universitetas" x="43" font="s" y="5" textcolor="##666666" />
        <label text="KTU simbolis" fontstyle="b" font="s" x="43" y="18"
textcolor="##000000" />
        <image source="/fun/a/b.png" x="142" y="35"/>
        <image source="/fun/a/s.png" x="150" y="35"/>
        <image source="/fun/a/s.png" x="157" y="35"/>
        <image source="/fun/a/r.png" x="165" y="34"/>
    </panel>
    <panel name="toptone" x="0" y="0" w="176" h="80" onclick="goto('/fun/t/1.xml');"
drawborder="false" backgroundimage="/teaserback_01.png"
selectedbackgroundimage="/teaserback_02.png" >
        <image source="/fun/t/1.png" x="5" y="5" />
        <label text="KTU Karjera" x="43" font="s" y="5" textcolor="##666666" />
        <label text="Karjeros dienos" fontstyle="b" font="s" x="43" y="18"
textcolor="##000000" />
        <image source="/fun/a/l.png" x="131" y="34"/>
        <image source="/fun/a/s.png" x="141" y="35"/>
        <image source="/fun/a/b.png" x="149" y="35"/>
        <image source="/fun/a/s.png" x="157" y="35"/>
        <image source="/fun/a/r.png" x="165" y="34"/>
    </panel>
    <panel name="topgame" x="0" y="0" w="176" h="80" onclick="goto('/fun/g/menu.xml');"
drawborder="false" backgroundimage="/teaserback_01.png"
selectedbackgroundimage="/teaserback_02.png" >
        <image source="/fun/g/news_p.png" x="5" y="5" />
        <label text="Naujienos" x="43" font="s" y="5" textcolor="##666666" />
        <label text="KTU ivykiai" fontstyle="b" font="s" x="43" y="18"
textcolor="##000000" />
        <image source="/fun/a/l.png" x="131" y="34"/>
        <image source="/fun/a/s.png" x="141" y="35"/>
        <image source="/fun/a/s.png" x="149" y="35"/>
        <image source="/fun/a/b.png" x="157" y="35"/>
    </panel>
</list>
<list name="menu" x="48" y="168">
    <button text="KTU Naujienos" backgroundimage="/fun/a/b0.png"
selectedbackgroundimage="/fun/a/bl.png" icon="/fun/w/wicon0.png"
selectedicon="/fun/w/wicon1.png" onclick="goto('/fun/g/menu.xml');" labelcolor="##000000"
selectedlabelcolor="##FF0000" />
    <button text="KTU Vaizdai" backgroundimage="/fun/a/b0.png"
selectedbackgroundimage="/fun/a/bl.png" icon="/fun/g/gicon0.png"
selectedicon="/fun/g/gicon1.png" onclick="goto('/fun/w/menu.xml');" labelcolor="##000000"
selectedlabelcolor="##FF0000" />
    <button text="KTU Karjera" backgroundimage="/fun/a/b0.png"
selectedbackgroundimage="/fun/a/bl.png" icon="/fun/t/ticon0.png"
selectedicon="/fun/t/ticon1.png" onclick="goto('/fun/t/menu.xml');" labelcolor="##000000"
selectedlabelcolor="##FF0000" />
</list>
<softkey keycode="back" name="Pabaiga" onclick="exit();" />

```

```
</window>
```

Failas Menu1.xml:

```
<window backgroundimage="/bg_new.png">
<image name="header" source="/header.png" x="0" y="7" focusable="false" />
<image name="header" source="/teaserback_wall.png" x="30" y="80" focusable="false"/>
  <label text="KTU Vaizdai" font="p" fontsize="s" fontstyle="b" x="40" y="80"
textcolor="#000000" />
  <list name="aabb" x="48" y="105" h="50" w="140">
    <button backgroundimage="/fun/w/g/1p.png"
selectedbackgroundimage="/fun/w/g/1p_s.png" focusable="true" onclick="goto('/fun/w/g/1.xml');"
selectedlabelcolor="#000000" />
    <button backgroundimage="/fun/w/g/2p.png"
selectedbackgroundimage="/fun/w/g/2p_s.png" focusable="true" onclick="goto('/fun/w/g/2.xml');"
selectedlabelcolor="#000000" />
    <button backgroundimage="/fun/w/g/3p.png"
selectedbackgroundimage="/fun/w/g/3p_s.png" focusable="true" onclick="goto('/fun/w/g/3.xml');"
selectedlabelcolor="#000000" />
    <button backgroundimage="/fun/w/g/3p.png"
selectedbackgroundimage="/fun/w/g/3p_s.png" focusable="true" onclick="goto('/fun/w/g/3.xml');"
selectedlabelcolor="#000000" />
    <button backgroundimage="/fun/w/g/3p.png"
selectedbackgroundimage="/fun/w/g/3p_s.png" focusable="true" onclick="goto('/fun/w/g/3.xml');"
selectedlabelcolor="#000000" />
    <button backgroundimage="/fun/w/g/3p.png"
selectedbackgroundimage="/fun/w/g/3p_s.png" focusable="true" onclick="goto('/fun/w/g/3.xml');"
selectedlabelcolor="#000000" />
  </list>
  <image source="/fun/a/1.png" x="35" y="115"/>
  <image source="/fun/a/r.png" x="190" y="115"/>
  <image name="header" source="/te.png" x="30" y="180" focusable="false"/>
  <label text="Kiti" font="p" fontsize="s" fontstyle="b" x="40" y="180"
textcolor="#000000" />
  <list name="menu" x="50" y="205" focusable="true" >
    <button backgroundimage="/fun/a/b0.png"
selectedbackgroundimage="/fun/a/b1.png" icon="/fun/w/wicon0.png"
selectedicon="/fun/w/wicon1.png" text="Fakultetai" labelcolor="#000000"
selectedlabelcolor="#FF0000"/>
    <button backgroundimage="/fun/a/b0.png"
selectedbackgroundimage="/fun/a/b1.png" icon="/fun/w/wicon0.png"
selectedicon="/fun/w/wicon1.png" text="Studijos" labelcolor="#000000"
selectedlabelcolor="#FF0000"/>
  </list>
  <softkey keycode="back" name="Atgal" onclick=" back();" />
</window>
```

Failas Menu2.xml

```
<window backgroundimage="/bg_new.png">
```

```

<image name="header" source="/header.png" x="0" y="7" focusable="false" />
  <image source="/teaserback_01.png" x="33" y="93"/>
  <label text="KTU karjera" fontstyle="b" font="s" x="90" y="95"
textcolor="#000000"/>
  <label text="Ateik!" fontstyle="b" font="s" x="90" y="107" textcolor="#000000"/>
  <image source="1.png" x="39" y="98"/>
  <label text="Reklama" x="90" y="120" textcolor="#000000"/>
  <softkey keycode="ok" name="Klausyti" onclick="
playFile('file:///fun/t/1.mp3','1');"/>
  <button x="49" y="187" text="Klausyti" backgroundimage="/fun/a/b0.png"
selectedbackgroundimage="/fun/a/bl.png" icon="/fun/t/ticon0.png"
selectedicon="/fun/t/ticon1.png" onclick="playFile('file:///fun/t/1.mp3','1');"
labelcolor="#000000" selectedlabelcolor="#FF0000" w="145" h="20"/>
  <softkey keycode="back" name="Atgal" onclick="back();" />
</window>

```

D Konferencijų medžiaga

EGNOS AUGMENTED GPS AND SISNET INTERFACE

Dominykas Barisas

Kauno technologijos universitetas

The aim of the investigation was to analyze EGNOS and SISNeT and apply correction data to GPS measurements based on C/A code, an interface of SISNeT, correction estimation. The improvement of accuracy of GPS raw data using different sources of EGNOS messages was analyzed and compared with each other and with the position from a commercial EGNOS receiver.

INTRODUCTION

Global Positioning System (GPS) is a satellite based navigation system designated to pinpoint any position on Earth or in a low orbit [1]. This system was developed by US military but later it was demilitarized. With passage of time and advancement of an infrastructure the precision which was provided by GPS became insufficient, therefore methods (DGPS, phase observation) and systems (SBAS) were invented to remove error sources such as tropospheric, ionospheric, satellite clock offset and satellite orbit errors. The basic principle of EGNOS is to transmit correction messages through the three geostationary satellites to the end-users so that they could remove errors from the GPS signals and achieve better accuracy [2]. This project will introduce one of the SBAS currently active in Europe - European Geostationary Navigation Overlay Service (EGNOS), its investigation, implementation of the Internet based SISNeT Interface and its analysis.

EGNOS AND SISNET

EGNOS definition

European Geostationary Navigation Overlay Service (EGNOS) is a wide area Satellite Based Augmentation System (SBAS) which broadcasts augmented GPS data over the whole continent and complements GPS and Russian Global Navigation Satellite System (GLONASS). Covering areas of the different systems are shown in Figure 0.1. It has been designed to broadcast a GPS-like ranging signal in Europe with embedded corrections, providing improved performances. A network of ground Ranging and Integrity Monitor Stations (RIMS) receive the corresponding satellites signals and transfer these to 4 central data processing stations. In these data processing stations a correctional map is calculated from the sum of the measurements of all monitor stations for total area. This correctional information is then sent to geostationary communication satellites, from where they are relayed to the GPS receivers. EGNOS improves the accuracy of GPS approximately from 5-12 meters to 1-to-3 meters [4][5]. Improvement of GPS performance comes with gathering the complementary data. It will operate on the GPS L1 frequency as well as standard GPS receivers. EGNOS improves performance, removing the need for local-area differential and commercial services in many cases. The project describes the EGNOS requirements and the overall system design.

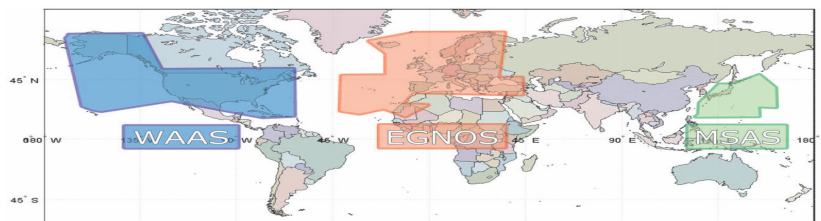


Figure 0.1. SBAS Systems.

EGNOS is the first step on the European contribution to the Global Navigation Satellite System, and a part of GALILEO, Europe's own Global Navigation Satellite System, which is going to be launched in 2010. It will give benefit to many different fields: maritime, railways, road transport, agriculture etc.

EGNOS consists of the four segments: ground segment, space segment, user segment and support facilities. Coverage areas of different GEOs are shown in Figure 0.2.

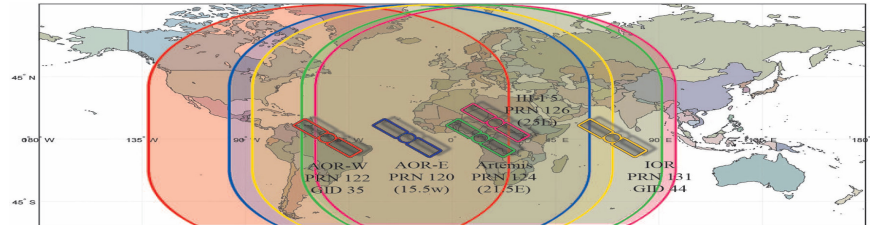


Figure 0.2. Coverage areas of GEO satellites.

SISNeT technology

The Signal in Space through the Internet (SISNeT) technology was launched in 2001 and provides access to the EGNOS messages through the Internet [3].

EGNOS signal may be computed without having an EGNOS receiver. These receivers are rather expensive and the GPS receiver together with the Internet connection may be used instead of it. There is a low requirements for the data transfer rate on the Internet (300 – 700 bps). This enables SISNeT usage with GSM/GPRS wireless networks. Additional information can be received with GSM/GPRS signals while GPS/GEO satellite is not available (garage, tunnel). Such helpful information as the last broadcasted messages or initialized GPS ephemeris information may be provided. SISNeT will open a large amount of applications for satellite navigation, which also may be used for educational purposes.

PROBLEM STATEMENT

Investigation of GPS position computation, EGNOS correction messages effect on GPS position and implementation of SISNeT interface are the basis for this project, which leads to the following problem statement:

- Computed position using raw data retrieved from GPS receiver with the correction messages applied
- Computed position using raw data retrieved from GPS receiver
- Computed position using raw data retrieved from GPS receiver with different corrections applied

European Space Agency (ESA) provides the EGNOS GPS augmentation services via geostationary satellites and the Internet. The objective of this project is to analyze the basic GPS and EGNOS concepts, implement, test, compare and evaluate different methods of GPS position calculation with and without EGNOS corrections applied.

IMPLEMENTATION

DS2DC protocol and commands

The DS2DC protocol gets the text strings composed of different fields separated by commas, which are transferred through TCP/IP protocol (between the DS2DC Client and the DS). The first field of such text command is an uppercase text and identifies the purpose of the command.

A few currently available DS2DC commands are provided:

MSG That is a request for EGNOS message. Transfer rate is limited to one command each second, otherwise the communication errors may occur. "*ERR" command will be received in case the DS is not able to answer.

AUTH The authentication process is necessary before starting receiving any EGNOS messages from the DS.

***MSG** It contains the answer to the EGNOS message. This message is transferred in hexadecimal format and is composed of 631 digits (250 bits for the message and the last two bits can be used by the users for detecting transmission errors).

EGNOS Message Server

EGNOS Message Server (EMS) is one of the ways to access EGNOS messages [11][13]. There are one hour old messages stored and the older ones may be accessed as well. EMS has very convenient structure of directories which enables user to find messages according to the PRN or time more easily. Each file contains one hour data, message is provided in hexadecimal format.

Structure and corrections

A brief structure of the system is shown in Figure 0.1. One of the major parts of this project is computation and application of the corrections. Ionospheric corrections are the most difficult ones and consist of three major parts. First of all, ionospheric pierce point is computed, then four or three nearest grid points depending on how many of them are available are chosen and interpolation algorithm applied. Eventually, the ionospheric correction IC_i to be added to pseudorange measurement is computed.

Fast corrections

The fast corrections are used for quickly changing errors such as satellite clock error. Fast correction is pseudorange correction (PRC) and is directly added to the pseudorange of every satellite. They are common to all users.

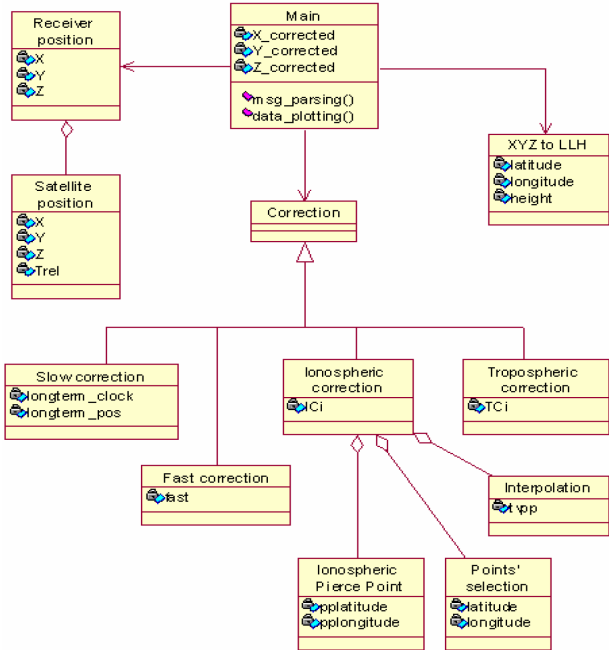


Figure 0.1. Brief structure of the system.

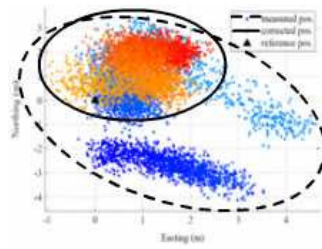


Figure 0.3 shows the influence of the fast correction, points are spread, but still within 3 meters level. It is so, because these errors change rapidly and they differs for each epoch. But it is rather small error mostly and position accuracy is improved much without using this correction.

The other error is tropospheric delay. The correction of it does not require any data from EGNOS messages and mostly depends on elevation angle of satellite. It changes together with satellite position change, influence of this correction is quite significant and is shown in Figure 0.5.

The correction illustrated in Figure 0.4 is only the long-term satellite clock correction, because satellite position error correction is equal to 0 and does not make any influence for the final position. This correction is fixed and does not change for a long time comparing to the others and mostly carries in only the arrangement of points at the different place without spreading them around. Furthermore, we even get a better precision in respect of Northing axis without applying this correction.

Ionospheric error causes the spread of the final position points as it is seen in **Figure 0.6**. This correction is estimated by computing a pierce point of each satellite and taking a vertical delay of it into account. When satellites are moving, the location of the pierce point changes and the error as well.

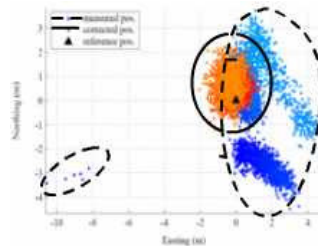


Figure 0.1. EGNOS corrected position (solid line – corrected positions, dashed line – measured position).

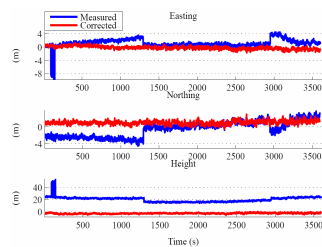


Figure 0.2. Coordinates before and after correction application.

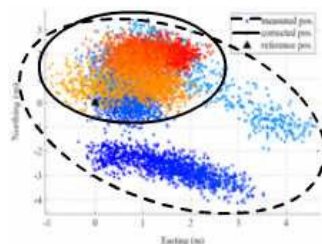


Figure 0.3. EGNOS corrected position without fast corrections applied (solid line – corrected positions, dashed line – measured position).

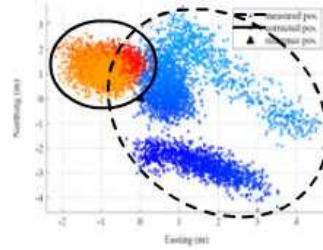


Figure 0.4. EGNOS corrected position without long-term corrections applied (solid line – corrected positions, dashed line – measured position).

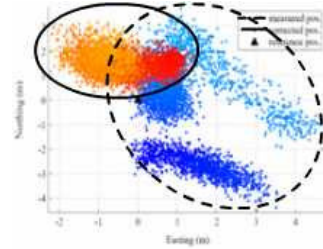


Figure 0.5. EGNOS corrected position without tropospheric corrections applied (solid line – corrected positions, dashed line – measured position).

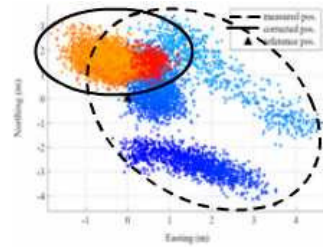


Figure 0.6. EGNOS corrected position without ionospheric corrections applied (solid line – corrected positions, dashed line – measured position).

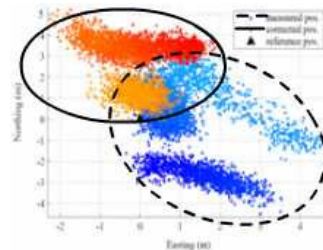


Figure 0.7. EGNOS corrected position without any corrections applied (solid line – corrected positions, dashed line – measured position).

DISCUSSION

Tests which show the influence of correction and the difference between different source of messages were made. However, much more testing should be done to solve some questions raised in the project like effect of evaluating correction messages accuracy or situations with limited number of satellites. This project focuses on two major parts: EGNOS correction

computation. Each of EGNOS corrections (fast, tropospheric, long-term, ionospheric) were computed and influence to the final position showed.

Even though EGNOS is still in testing phase its effectiveness is close to ESA statements. During test it was observed some points that were deviated from position up to eight meters, but comparing it to the total amount of collected data they are only a small part and they were not taken into consideration.

During tests time SISNeT was under testing. Some inadequacies were faced, firstly some empty messages were received which contain the right prefix but data fields are empty, moreover correction messages MT 2, 6 and 24 were never received, when messages are available. Correction messages MT 2 and 24 are needed to apply fast corrections for the first 13 satellites, therefore SISNeT loses a part of its functionality.

CONCLUSION

The following data was acquired:

- Position from EGNOS receiver with the SBAS mode (corrected position) turned on.
- GPS receiver measurements based on C/A code: satellite numbers, pseudoranges, elevation and azimuth angles were retrieved.
- EGNOS correction messages from the SISNeT, EGNOS receiver and FTP directory.

System tests were performed successfully and results collected. One of the University GPS antennas was used which position is known very accurately. All the main goals defined at the start of the project were fulfilled. Messages through the SISNeT were received and EGNOS correction data applied. After EGNOS corrections computation and application, the position accuracy and precision were improved, most of the achieved results were satisfactory and within the limits defined for EGNOS.

BIBLIOGRAPHY

- [1] **G. Strang, and K. Borre**, Linear Algebra, Geodesy, and GPS, 2nd ed., Wellesley-Cambridge Press (1997), ISBN 0-9614088-6-3.
- [2] **F. Toran-Marti, and J. Ventura-Traveset**, The First Step of the European Contribution to the Global Navigation Satellite System (GNSS), European Space Agency, Toulouse (France) (2004). GNSS-1 Project Office.
- [3] **F. Torán-Martí, D. J. Ventura-Traveset, and D. R. Chen**, "The ESA SISNeT Technology: Real-Time Access to the EGNOS Services through Wireless Networks and the Internet," ION GPS 2002 (September 2002), pp. 866–870. Portland, OR.
- [4] **RTCA**, Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment, RTCA (2001). DO-229C.
- [5] **M. Hernandez-Pajares, J. M. Juan, J. Sanz, and X. Prats**, EGNOS Tutorial, University Politecnica de Catalunya (2002). Research group of Astronomy and GEomatics(gAGE/UPC).
- [6] **N. Penna, A. Dodson, and W. Chen**, "EGNOS Tropospheric Correction Model," Assessment of EGNOS Tropospheric Correction Model, pp. 39–41. IESSG, University of Nottingham.
- [7] **G. T. Team**, PEGASUS Technical Notes on SBAS, Eurocontrol (2003). Project: PEGASUS.
- [8] **P. Høeg, G. B. Larsen, H.-H. Benzon, S. Syndergaard, and M. D. Mortensen**, "The GPSOS project," Algorithm Functional Design and Analysis of Ionosphere, Stratosphere and Troposphere Observations 98-2, Danish Meteorological Institute, Solar-Terrestrial Physics Division, Lyngbyvej 100, DK-2100 Copenhagen Ø, Denmark.
- [9] **F. Halsall**, Multimedia Communications, Pearson Education Limited (2001), ISBN 0-201-39818-4.
- [10] **F. Toran, and D. J. Ventura-Traveset**, SISNeT User Interface Document, European Space Agency, Toulouse (France), 2nd ed. (2002May). GNSS-1 Project Office.
- [11] **F. Toran-Marti, and J. Ventura-Traveset**, EGNOS Message Server (EMS) User Interface Document, European Space Agency, Toulouse (France) (2004 May). GNSS-1 Project Office.
- [12] **Sam**, "GPS Raw Data," URL <http://home-2.worldonline.nl/~samsvl/>.
- [13] **E. S. A. (ESA)**, "EGNOS messages," URL <ftp://ems.estec.esa.int/pub/>.

EGNOS patobulinta GPS ir SISNeT Interfeisas

Šio projekto tikslas yra gilesnė EGNOS ir SISNeT analizė, SISNeT interfeiso sukūrimas ir informacijos apie korekcijas pritaikymas matavimams paremtiems GPS imtuvo C/A kodu bei testavimas. Projekte naudojami prietaisai yra komercinė GPS antena, EGNOS ir GPS imtuvai prijungti prie kompiuterio per nuoseklias jungtis, kurių pagalba į imtuvą siunčiama užklausa ir gaunami duomenys atitinkamais laiko intervalais naudojant Matlab programą. Iš GPS imtuvo gautų pozicijų tikslumai gerinami naudojant skirtingus EGNOS žinučių šaltinius (iš EGNOS imtuvo, SISNeT interfeiso, FTP direktorijos), palyginti tarpusavyje ir su duomenimis iš komercinio EGNOS imtuvo bei pateiktos išvados.