

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PRAKTINĖS INFORMATIKOS KATEDRA

DARIUS BAKŠYS

**MOKYMO INFORMACIJOS TRANSFORMAVIMAS IR  
ATVAIZDAVIMAS MOBILIUOSE ĮTAISUOSE**

Magistro darbas

Darbo vadovas  
prof. V. Štuikys

KAUNAS, 2007

## TURINYS

Summary.....	5
1. Įvadas.....	6
2. Analitinė dalis.....	7
2.1. Mokymo(si) formos.....	7
2.1.1. Tradicinė pamoka.....	7
2.1.2. Elektroninis mokymas(is).....	8
2.1.3. Mobilus mokymas(is).....	9
2.2. DRESS metodas.....	13
2.2.1. Žiniatinklio puslapių transformavimo būdai.....	13
2.3. Dokumento atvaizdavimas naudojant kintamo dydžio struktūrą.....	14
2.4. Padalijimo medžio struktūra.....	15
2.4.1. Svarbumo reikšmė (IMP).....	15
2.4.2. Mažiausio matomo simbolio dydis/aukštis/plotis (MPS, MPA ir MPP).....	15
2.4.3. Pritaikomumas (ADJ).....	16
2.4.4. Pasirinkimas.....	16
2.5. Automatinis DRESS generavimas.....	16
2.6. Padalijimų medžio generavimas.....	17
2.6.1. Informacijos svarbumo reikšmė.....	18
2.6.2. Kiti atributai.....	18
2.7. DRESS pagrindu sukurtų dokumentų pritaikymas.....	18
2.7.1. Informacijos tikslumas.....	19
2.7.2. Problemos apibrėžimas.....	19
2.7.3. Puslapio atvaizdavimo (rendering) algoritmas.....	20
2.7.4. Bloko rinkinio pasirinkimas.....	20
2.7.5. Talpos padalijimas.....	21
2.7.6. Turinio pritaikymas.....	23
2.8. DRESS metodo realizacija mobiliuose įtaisuose.....	24
2.8.1. DRESS realizacija naršyklėse.....	24
2.8.2. Algoritmo patikrinimas.....	24
3. Projektinė dalis.....	31
3.1. Architektūros tikslai ir apribojimai.....	31
3.1.1. Apribojimai sprendimui.....	33
3.1.2. Diegimo aplinka.....	33
3.1.3. Bendradarbiaujančios sistemos.....	33
3.1.4. Komerciniai specializuoti programų paketai.....	33
3.1.5. Numatoma darbo vietos aplinka.....	34
3.2. Praktikos metu taikytos testavimo strategija:.....	34
3.2.1. Vienetų testavimas.....	34
3.2.2. Integravimo testavimas.....	34
3.2.3. Priėmimo testavimas.....	34
3.2.4. Aukšto lygio testavimas.....	35
3.2.4.3. Testavimo resursai.....	35
3.2.4.4. Testavimo įrankiai ir aplinka.....	35
4. Tyrimo dalis.....	36
4.1. Svarbios informacijos kodavimas md5 algoritmu.....	36
4.2. Greitas rezultatų pateikimas pagal paieškos raktus.....	37
4.3. Perėjimo tarp puslapių efektyvinimas.....	37
4.4. El.pašto kodavimas wml programavimo kalba.....	38
5. Eksperimentinė dalis.....	39
6. Išvados.....	42

7. Literatūra .....	43
8. Terminų ir santrumpų žodynas .....	45
9. Priedai .....	46

## SANTRAUKA

Darbe plačiau gvildenamas vienas iš galimų mobilaus mokymosi informacijos gavimo būdų – tai elektroninio mokymosi informacijos transformavimas ir adaptavimas mobiliuose įtaisuose, atsižvelgiant į juos įtakojančius apribojimus. Apibrėžiami transformavimo aspektai, galinčios kilti problemos, pateikiami jų sprendimo būdai – naudojamas DRESS metodas bei šakų-ribų algoritmas. Darbe atliktas žiniatinklio puslapių kūrimo rankiniu ir automatiniu būdų palyginimas, rodantis kad antrasis būdas leidžia sutaupyti laiko, nereikia mokytis naujos programavimo kalbos, informacijos nereikia naujai suvedinėti į mokymosi. Darbas apibendrinamas pateikiant užsienio tyrinėtojų atlikto informacijos apdorojimo glaustumo (thumbnails) ir DRESS metodai rezultatus. Pabaigoje sukurtas magistrinio projekto dalies transformavimo modulis, atsižvelgiant į išnagrinėtus transformavimo žingsnius.

## SUMMARY

In this work is speaking about learning information transformation and adaptation for mobile device, define transformation aspect, problems, which can be, giving solution way – is using DRESS method and branch-bound algorithm.

In work is giving web page creation by manual and automatic way comparison. Automatic learning information transformation way let for us save time, escape new programming language studying, information don't want bring into learning server – it can be receive from e.learning server. The work is summarize giving information adaptation result, where is comparing thumbails and DRESS method and which make foreign researcher. In finally, giving project transformation model, which was created by transformations steps.

## 1. ĮVADAS

Mobilūs įtaisai, tokie kaip mobilūs telefonai, asmeniniai skaitmeniniai pagalbininkai (PDA) ir nešiojami kompiuteriai pastaruoju metu tapo labai populiarūs. Šie įtaisai leidžia mobiliai bendrauti, lengvai pasiekti informaciją bet kuriuo laiku ir bet kada. Sukurtos mobilios technologijos ima vis labiau skverbtis ir į mokymą(si). Atsiranda nauja paradigma, vadinama mobiliu mokymusi – tai mokymas(is) judant, keliaujant. Ateityje tikimasi, kad ši mokymo(si) veikla pamažu persikels iš įprastos klasės į mokinio aplinką [18]. Tai mokymą(si) padarys labiau bendru, visą gyvenimą trunkančiu procesu. Mobilaus mokymo(si) tyrinėjimai vystomi. Pasauliniu mastu vykdyta keletas didesnių iniciatyvų (pvz., MOBILEarn[20], mlearning[19]), kurių metu buvo ištirta galima šios naujai plintančio mokymo(si) formos nauda. Mobilaus mokymo(si) platesnį pritaikymą stabdo eilė socialinių ir technologinių apribojimų, tokių kaip telekomunikacijų išlaidos [18]; mobilaus įtaiso apribojimai - maža procesoriaus galia (išskyrus nešiojamus kompiuterius), maža atminties talpa, mažas ekrano dydis, nedidelis belaidžio tinklo pralaidumas [21, 22].

Kita svarbi kliūtis mobilaus mokymo(si) taikymui yra mobilaus mokymosi programų sudėtingas plėtojimas - nėra sukurta efektyvių vystymo įrankių. Kadangi mobilus mokymas(is) glaudžiai susijęs su elektroniniu mokymusi, tai jo medžiagą, vykdant mokymo(si) informacijos transformavimą automatinio arba kuriant mobilaus interneto puslapius rankiniu būdu, naudojant wml programavimo kalbą, galima pateikti tokia forma, kurią būtų galima sėkmingai atvaizduoti mobilių įtaisų ekranuose.

## 2. ANALITINĖ DALIS

Šioje dalyje kalbama apie darbo objektą – mokymą(si), jo formas, tokias kaip tradicinę pamoką, elektroninį mokymą(si), tačiau labiausiai akcentuojamos mobilus mokymo(si) galimybės, informacijos transformavimo ir adaptavimo mobiliuose įtaisuose aspektai, pedagoginiai ir technologiniai apribojimai, sąryšiai su naujomis technologijomis.

Literatūros apžvalga sudaryta iš trijų dalių:

I dalis – Pedagoginiai aspektai.

II dalis – Technologiniai mokymo(si) informacijos transformavimo ir adaptavimo aspektai.

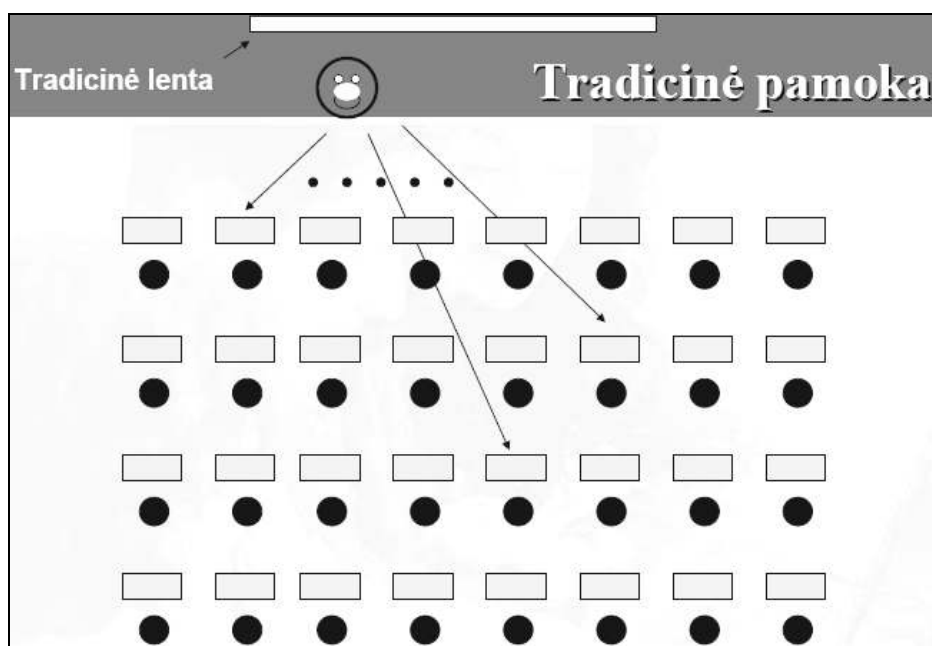
III dalis – Magistrinio darbo projekto dalies transformavimo modelis.

*I dalis – Pedagoginiai aspektai*

### 2.1. Mokymo(si) formos

Dabartinėje švietimo sistemoje egzistuoja trys pagrindinės mokymo(si) formos: tradicinė pamoka, elektroninis mokymas(is), mobilus mokymas(is). Šios formos tarpusavyje glaudžiai susijusios, viena kitą papildo. Norint plačiau suprasti mobilų mokymą(si) reikia nagrinėti elektroninį mokymą(si), o jo atžvilgiu reikia gerai suprasti ir žinoti tradicinės pamokos struktūrą.

#### 2.1.1. Tradicinė pamoka



1 pav. Tradicinės pamokos grafinis atvaizdavimas [23]

Tradicinė pamoka - tai mokymo(si) forma, kai mokytojas atsistojęs visų akivaizdoje aiškina pateikdamas informaciją lentoje, o mokinys klauso, rašo ir klausia. Suolų išdėstymas dažniausiai būna tradicinis – eilėmis, taip, kad kiekvienas mokinys gerai matytų mokytoją ir jo lentoje pateikiamą informaciją. Tradicinėje pamokoje labai svarbus išlieka žmogiškasis bendravimas, kai mokytojas pakankamai greitai gali atsakyti į mokinio klausimus. Tačiau tokioje pamokoje mokiniai turi pateikiamą informaciją sugebėti užsirašyti pakankamai greitai neatsilikdami nuo kitų. Pamokos vedimo tvarką kontroliuoja mokytojas, atsižvelgdamas į pamokos struktūrą bei įvertindamas esamą situaciją pamokoje (esančių mokinių skaičių, pateikiamos informacijos sunkumą). Tai dar plačiai švietimo sistemoje naudojama mokymo forma, nors daugelio daug kritikuojama. Tačiau ji dar gyvuoja, nes mokymas(is) vyksta tiesiogiai bendraujant proceso dalyviams (mokytojui ir mokiniui).

### **2.1.2. Elektroninis mokymas(is)**

Elektroninis mokymas(is) (toliau *el. mokymas(is)*) – tai mokymas(is) elektroniniu būdu. El. mokymas(is) yra sąveikaujantis, tinkluose vykstantis mokymas(is), čia didelis dėmesys skiriamas mokymo(si) turiniui, o ne tarpusavyje bendraujantiems mokymo(si) dalyviams. El. mokymą(si) galima laikyti mokymusi naudojantis kompiuteriu (computer – based training), mokymusi virtualiose klasėse (virtual classroom), kur mokinys - mokytojas, mokinys - mokinys arba mokytojas - mokytojas bendrauja tarpusavyje, dalijasi informacija, dalyvauja forumo diskusijose ir t.t. Mokymo(si) medžiaga pateikiama ir pasiekama internetu, intranetu, ektranetu, naudojantis audio ir video laikmenomis [26].

El. mokymo(si) teikiami privalumai:

- Galimybė mokiniams mokytis bet kur ir bet kada, kur yra internetinė prieiga.
- Galimybė įsisavinti pateiktą medžiagą mokiniui priimtiniu mokymo(si) tempu bei laiku.
- Galimybė patirti mažesnę stresą atliekant užduotis; tiesioginis bendravimas su mokytoju bei klasės draugais dalyvaujant forumose.

Sėkmingam el. mokymui reikia:

- Kurti ir plėtoti mokymuisi reikalingas el.mokymo programas.
- Nuolat atnaujinti kompiuterinę ir programinę įrangą.
- Kad mokymo planavime dalyvautų IT personalas.



### 2.1.3. Mobilus mokymas(is)

Mobilus mokymas(is) (toliau *mob. mokymas(is)*) – tai mokymas(is), naudojant mažus, nešiojamus informacijos apdorojimo (computing) įtaisus - sumanius telefonus (smartphones), asmeninius skaitmeninius pagalbininkus (personal digital assistants (PDAs)), delninius, nešiojamus kompiuterius, kurie naudojami keliaujant, judant. Tai galimybė mokytis bet kur ir bet kada [25].

*Mob. mokymo(si) charakteristikos:*

- Mokymo(si) reikalingumo būtinumas
- Žinių įsigijimo iniciatyva
- Mokymo aplinkos mobilumas
- Mokymo proceso sąveikavimas
- Mokomųjų veiklų išdėstymas
- Mokymo turinio integracija

Šios charakteristikos gali sudaryti mokymo teorijų specifikas.

Mokymas(is) yra visą gyvenimą besitęsiantis procesas, susidedantis iš situacijų. Mob. mokymas(is) yra pirmoji technologija, kuri pilnai integravusi kasdienines veiklas, kad palaikytų visą gyvenimą besitęsiantį mokymo procesą, kaip galima matyti pateikiamoje žemiau lentelėje, nesvarbu, kokia pedagogika yra naudojama.

1 lentelė. Mokymo(si) visą gyvenimą ir mob.mokymo(si) susiejimas[27]

<b>Visą gyvenimą besitęsiančio mokymo(si) atributai</b>	<b>Mob. mokymo(si) atributai</b>
Individualizavimas (smulkus apibūdinimas)	Asmeninis
Koncentracija į mokinį	Koncentracija į vartotoją
Situacijų valdymas	Mobilumas
Bendradarbiavimas	Tinklinis
Visur esantis	Visur esantis
Visą gyvenimą trunkantis	Ilgaamžis

Ne mažiau svarbūs m.mokymosi pritaikymo procese yra ir technologiniai apribojimai.

*Mobiliosios technologijos ir mobiliųjų įtaisų apribojimai*

Mobiliosios technologijos - tai technologijos, leidžiančios informaciją gauti nepriklausomai nuo buvimo vietos. Informacija gaunama naudojant tokius mobilius įtaisus kaip nešiojamieji ir PDA kompiuteriai, mobilūs telefonai ir pranešimų gavikliai.

Mobilusis prietaisas – mažas mobilus įtaisas, kuriame galima saugoti, gauti bei valdyti gaunamą informaciją. Mobilieji prietaisai gali dirbti naudodami skirtingas operacines sistemas (*Microsoft Windows mobiliems įtaisams* arba *Linux*). Mobiliuose įtaisuose simboliai įvedami sumažinta klaviatūra arba naudojant specialų pieštuką (*Stilus*). Įtaisai naudojam pakraunamus elementus. Įtaisas gali būti vizitinės kortelės ar net knygos dydžio.

Populiariausi mobilieji įtaisai:

- PDA kompiuteriai.
- Sumanieji (smart) telefonai;

Sumanusis telefonas – tai rankinis įtaisas, jungiantis mobiliojo telefono ir PDA kompiuterio galimybes. Jam suteikiamos sumanumo savybės.

*PDA kompiuteris* – tai mobilusis įtaisas, kurio pagrindinė funkcija - asmeninės informacijos valdymas. Tobulėjant informacinėms technologijoms, PDA kompiuteriai tapo galingais mobiliaisiais ir bevieliais įtaisais. Yra PDA kompiuterių, kurie naudoja sinchronizavimo programas (informacija juose atnaujinama prijungus PDA kompiuterį prie stalinio kompiuterio). Naujesni prietaisai naudoja belaidį ryšį ir duomenys tokiu būdu atnaujinami iš karto.

Bevielių tinklų, įrenginių technologijos ir tiems įrenginiams pritaikytų paslaugų platus pritaikymas sąlygojo tai, kad, jog visame pasaulyje žymai išaugo mobiliųjų technologijų naudojimas, kas leido ir įmonėms, ir paprastam vartotojui naudojantis mobiliaisiais įtaisais jungtis prie interneto bet kuriuo laiku ir iš bet kur. Mobilias programas galima pritaikyti plačiai - viešbučių rezervavimui kelionės metu, kelio paieškai vairuojant, el. pašto tikrinimui ir pan.

Sukurta GPRS technologija leidžia naudojantis GSM mobiliu tinklu naršyti internete, gauti elektroninį laišką, saugiai prisijungti prie savo banko sąskaitos ir t.t. Siūlomas 100 kbps yra perpus didesnis nei naudojantis fiksuotojo telefono linija, tačiau realiai tokiu greičiu gali naudotis tik pačius naujausius GSM mobilius įtaisus turintys vartotojai.

*WAP (Wireless Application Protocol)* leidžia mobiliajame telefone skaityti elektrinius laiškus bei naršyti internete. WAP – tai bevielis duomenų perdavimo protokolas, kurį naudojama bevieliai įtaisai tarpusavio komunikacijai.

Plačiausiai WAP pritaikomas mobiliame internete – GSM ryšio ir interneto hibride. WAP naudojami puslapiai privalo būti parašyti WML programavimo kalba – tai supaprastinta HTML kalbos versija. Pyslapiai turi būti paprasti, kad informacija būtų greitai perduodama ir atvaizduojama mobiliame įtaise [3].

## *WAP ir WML programavimo kalba*

WAP (*Wireless Application Protocol* – bevielio ryšio programų protokolas) – tai ryšio protokolas, kuris aprašo, kaip vykdyti interneto duomenų perdavimą bevielio interneto ryšio kanalais.

WML (*Wireless Markup Language*) – tai programavimo kalba, HTML atšaka, naudojama tvarkant interneto duomenis ir leidžianti juos atvaizduoti mobiliųjų įtaisų, palaikančiuose WAP, ekranuose.

Mobilieji įtaisai gali priimti ir išsiųsti duomenis 9,6 Kbps sparta, o tai labai lėta. Tokios yra GSM standarto galimybės, tačiau sukurti nauji standartai, tokie kaip GPRS, UMTS leidžia šią spartą padidinti dešimtimis kartų [3].

Tam, kad WAP galėtų gerai veikti, reikia atkreipti dėmesį į šiuos dalykus: siaura bevielių tinklų dažnių juosta bei specifinės tinklų terminalų ypatybės, tokios kaip: maži ekranai ar nepakankama įrangoje instaliuota CPU galia. WML ir WMLScript - tai dvi naujos programavimo kalbos, kurios buvo sukurtos įvertinant aukščiau minėtus ribojančius dalykus [3].

Be minėtų apribojimų galime išskirti šiuos:

- Atminties talpa – ribotas informacijos saugojimas ir priėmimas mobiliame įtaise.
- Procesoriaus galia – nedidelė procesoriaus galia, galimybė apdoroti informaciją dalimis.

## *Kaip WML yra sudaryta?*

WML sudaro XML objektai, vadinami elementais. Svarbiausi elementai šie: korta ("card") ir kortų malka ("deck"). WML malką sudaro dvi kortos, į kurias vartotojas gali įsirašyti savo žodį - Pasisveikinimas (I-oje kortoje) bei sakinį "Sveikas Pasauly" (II-oje kortoje). Galima keisti ekrane rodomo teksto formatą.

Korta - mažiausias elementas, perduodamas tarp tinklo serverio ir kliento, o visa tai, kas yra rodoma ekrane, yra kortos turinys. Malką gali sudaryti viena arba daugiau kortų ir klientas gali peržiūrinėti įvairias kortas bei malkas. Reikia žinoti, kad trys pirmosios WML dokumento eilutės turi žymėti XML deklaraciją bei nuorodas į WML-DTD.

Magistrinio projekto metu kurta programinė įranga buvo realizuota rankiniu būdu, būtent programuojant žiniatinklio puslapius WML programavimo kalba.

*WML puslapio kodo pavyzdys:*

```
<wml>

<card id="card1" title="Tutorial">
<p>
Pasveikinimas
</p>
</card>

<card id="card2" title="Tutorial1">
<p>
Sveikas Pasauli!
</p>
</card>
</wml>
```

Rezultatas – rodomas tekstas mobiliame įtaise



2 pav. Rezultatas mobiliame telefone

## *II dalis – technologiniai mokymo(si) informacijos transformavimo ir atvaizdavimo mobiliuose įtaisuose*

### Tikslas

Pateikti ir formaliai aprašyti mokymo(si) informacijos transformavimo ir atvaizdavimo aspektus.

### Uždaviniai

- Nustatyti žingsnius, reikalingus atlikti automatinį mokymo(si) transformavimą.
- Pateikti formalų transformavimo aprašymą.
- Pateikti atvaizdavimo priklausomybę tarp mobilaus įtaiso ekrano ir suformuoto informacijos bloko.
- Pateikti magistrinio darbo projekto dalies transformavimo modelį.

Mobilūs įtaisai negali korektiškai ir pilnai atvaizduoti visos žiniatinklio medžiagos, nes tai riboja jų ekrano dydis, tinklo pralaidumas, maža atminties talpa. Todėl reikalinga vykdyti žiniatinklio puslapių semantinę analizę – informacijos skaidymą į informacijos blokus, kurie būtų atvaizduojami mobiliame įtaise. Transformavimas bus vykdomas naudojant DRESS metodą bei šakų – ribų, godų algoritmus.

## 2.2. DRESS metodas

Kadangi mobiliems įtaisams taikoma eilė apribojimų ir žinant, kad visos žiniatinklio informacijos neįmanoma vienu metu pateikti mobiliame telefone, tenka ieškoti būdų, kaip žiniatinklio informaciją suskaidyti į blokus bei ją atvaizduoti ekrane, atsižvelgiant į jam keliamus reikalavimus.

Šiai problemai spręsti galima pasitelkti DRESS technologiją. Taikant šį metodą, ekrano vieta yra efektyviau pritaikoma prie pateikiamos informacijos, o tai leidžia sėkmingai ekrane pateikti informaciją dalimis bei ją vėliau apjungti. Ši technologija taikoma dabartiniams žiniatinklio dokumento formatams, kurie ateityje bus labiau tobulinami.

### 2.2.1. Žiniatinklio puslapių transformavimo būdai

Pirmas - naudoti raktus, padedančius informacijos paieškos procese. Tai pagreintų pageidaujamos informacijos pateikimą vartotojui [5].

Antras - vykdyti žiniatinklio puslapio transformaciją – išskirti puslapio semantinę struktūrą ir pateikti puslapį pagal vartotojo reikalavimus.

Pagrindinė idėja – puslapį suskaidyti į subpuslapius, sugeneruoti turinio lentelę su hierarchija arba be jos. Galimas dviejų aukščiau minėtų technikų apjungimas ir naudojimas kartu [6,7]. Nors didelės pastangos dedamos į automatinį dokumento struktūros gavimą (extraction), tačiau vis dar yra sunku kompiuteriais efektyviai išskirti žiniatinklio puslapio semantinę struktūrą.

Žiniatinklio puslapių transformavimo procese vis labiau imta panaudoti straipsnio autoriaus žymes, sutartiniai – standartizuoti ženklai, kurie tiesiogiai gali padėti transformavimo fazėje. Transformavimo procesą palengvintų ir kitokio failo formatų naudojimas (XML ir XSLT) [8].

Koncentruosiuosi į kintančio dydžio nustatymo problemą ir efektyvią žiniatinklio dokumento struktūros išskyrimą – kas leis vėliau pateikiamą informaciją pritaikyti skirtingo dydžio ekranams.

Darbe bus naudojami dvejetainiai medžiai (binary slicing trees), žiniatinklio sluoksnių problema bus formuluojama kaip 0-1 kuprinės (0–1 knapsack) variantų problema, kuri gali būti sėkmingai išspręsta naudojantis šakų ir ribų algoritmu arba godaus algoritmo[11].

### 2.3. Dokumento atvaizdavimas naudojant kintamo dydžio struktūrą

Pradžioje apibrėškime terminą – DRESS, žiniatinklio struktūros išskyrimo ir adaptavimo metodą.

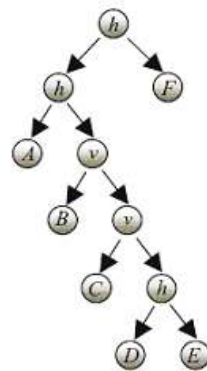
**Apibrėžimas.** DRESS – tai dvejetainis padalijimo (slicing) medis su  $N$  viršūnių. Medžio vidinis mazgas gaunamas taikant vertikalių ( $v$ ) arba horizontalių ( $h$ ) skaidymo būdą, tokiu būdu kiekvienas lapo mazgas yra informacinis blokas apibrėžiamas taip:

$$B_i = (IMP_i, MPS_i, MPA_i, MPP_i, ADJ_i, ALTi) \quad (1)$$

kur  $i = 1, 2, \dots, N$ ,  $B_i$  –  $i$ -tasis informacinis blokas žiniatinklio puslapyje (žr. 4 pav);  $IMP_i$  –  $B_i$  informacijos svarbumo parametras;  $MPS_i$  – mažiausias dar matomas simbolio dydis bloke;  $MPA_i$  – mažiausias simbolio aukštis;  $MPP_i$  – mažiausias simbolio plotis;  $ADJ_i$  – koeficientas, pagal kurį sprendžiama, ar blokas tilps mob. įtaiso ekrane;  $ALTi$  –  $B_i$  alternatyva.



(a)



(b)

3 pav. Žiniatinklio puslapio pavyzdys (a) ir jo padalijimo (slicing) medis (b) [16]

## 2.4. Padalijimo medžio struktūra

Pagal apibrėžimą, DRESS yra padalijimo medis su kiekvienu lapo mazgu, kuris yra žiniatinklio puslapio informacijos blokas. Vidinio mazgo kortelėje (label) nurodoma reikšmė, rodanti, kiek ekrano dalies rekursyviai yra padalijama į sub-stačiakampius pagal padalijimo medį, naudojant vertikalų arba horizontalų būdus. Informacijos blokai bus talpinami į suformuotų sub-stačiakampių vietą pagal lapo mazgą.

Žiniatinklio puslapio pavyzdys ir jo padalijimo medis parodyti 3 pav. Tai ebay svetainės puslapis ([www.ebay.com](http://www.ebay.com)).

Padalijimo medžio nauda ta, kad atvaizduoja – ketinamą loginę puslapio struktūrą, kurią sudaro skirtingi dideli blokai, kurie logiškai susiję, tokie kaip antraštė, pagrindiniai skyriai, šoninė juosta ir t.t. Tokiu būdu mes turime blokus, kurie dar vadinami informacijos blokais (Bi), kurie turi 6 atributus:

- Svarbumo reikšmė (IMP)
- Mažiausio matomo simbolio dydis (MPS)
- Mažiausio matomo simbolio aukštis (MPA)
- Mažiausio matomo simbolio plotis (MPP)
- Pritaikomumas (ADJ)
- Pasirinkimas (ALT)

### 2.4.1. Svarbumo reikšmė (IMP)

Kadangi skirtingi blokai talpina skirtingus informacijos kiekius ir turi skirtingas funkcijas, jie yra skirtingo svarbumo. Tokiu būdu, svarbumo reikšmę galime laikyti matu nustatant bloko svorį. Informacijos blokai išdėliojami svarbumo mažėjimo tvarka ir atitinkama tvarka atvaizduojami ekrane. Didžiausio svorio blokas pateikiamas geriausiai vartotojo matomoje zonoje, mažiau svarbūs (reklama ir pan), nukeliami į šalį. Ši reikšmė yra naudojama tada, kada yra naudojami mažiausiai svarbūs blokai, juos pritaikant prie mažų ekranų.

### 2.4.2. Mažiausio matomo simbolio dydis/aukštis/plotis (MPS, MPA ir MPP)

Šie atributai labai svarbūs sluoksnių optimizavimo procese. Dokumentų pateikimui mes galime panaudoti įvairias technikas, įvertinant tokius veiksmus, kaip teksto didinimas, šrifto keitimas, antraštės iškirpimas. Jeigu informacijos bloko turinys bus mažinamas, tai jis gali būti tapti nebeįskaitomas. Todėl ir reikalinga apibrėžti mažiausio galimo atvaizduoti simbolio dydį, aukštį, plotį.

Pavyzdžiui, tarkime turime trumpos naujienos informacijos bloką, kurio originalus srities dydis 30 000 taškų. Autorius arba leidėjas gali apibrėžti jo MPS, kuris gali būti 4 kartus mažesnis, t.y 7 500 taškų, tai dar leis tekstą perskaityti, nes teksto simbolis bus 9 taškų.

MPS, MPA ir MPP reikšmės priklauso nuo naudojamo konteksto, vartotojo regėjimo lauko, jo atstumo iki ekrano (pavyzdžiui, jeigu žiniatinklio puslapis rodomas televizoriuje, simboliai bus didesnis).

### 2.4.3. Pritaikomumas (ADJ)

Parametras, rodantis ar informacijos bloką bus galima atvaizduoti ekrane. Pavyzdžiui, jeigu turinio bloką sudaro grynas tekstas arba tekstas su piešiniu, jis gali būti apgaubtas ir tokiu būdu efektyviau pritaikytas atvaizdavime ekrane.

### 2.4.4. Pasirinkimas

Mažiausiai svarbiausias informacijos bloko turinys yra reklama, tokiu būdu pageidaujama, kad ji būtų apibendrinta, nes daugiau vietos liktų svarbiems informacijos blokams. Taip elgiamasi ir su dideliais blokais, kurie negali būti parodyti be pernelyg didelio sumažinimo iki riboto displejaus dydžio, tokiu būdu yra formuojama santrauka su nuoroda į originalų turinį. Vartotojas pasirinks, ar nori matyti informaciją dalimis ar matyti visą, peržvelgdamas ją naudojamas slinties juostos. Kaip alternatyva dažniausiai naudojamas trumpas tekstas, kuris trumpai nusako originalaus turinio mintį. Jo funkcijos yra panašios į HTML žymės IMG atributo ALT [12].

## 2.5. Automatinis DRESS generavimas

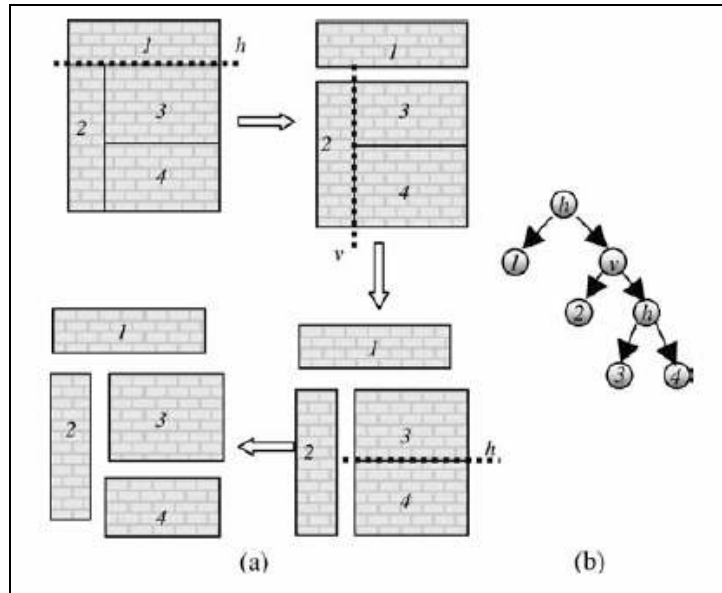
Padalijimo medžis (žr. 3 pav.) buvo formuojamas rankiniu būdu, tačiau jis gali būti sukurtas automatiškai, naudojant žiniatinklio puslapių struktūros analizę.

Generavimo esmė: informacijos blokų aptikimas žiniatinklio puslapyje. Anksčiau buvo kalbėta, kad turi būti vykdoma žiniatinklio puslapyje struktūros analizė - suskaidymas į smulkius ir logiškai susijusius vienetus, kurie priderinami prie mobilaus įtaiso ekrano [13].

Puslapio analizės algoritmas vykdomas 3 žingsniais:

Pirmas, HTML DOM medis yra analizuojamas ir aukščiausio lygio turinio blokai (antraštės, šoninės juostos, kūnas) yra identifikuojamos. Tada turinio viduje, kiekvienas aukšto lygio turinio blokas yra analizuojamas, kad būtų identifikuoti aiškūs skyrikliai suskaldantys turinio bloką. Pabaigoje, numanomi skyrikliai yra identifikuojami naudojant projektavimo algoritmą ir naudojami turinio blokų suskaidymui toliau.





4 pav. Skyrinklių (separator) paieškos algoritmas (a) ir galutinis padalijimo medis (b) [16]

Čia panaudojome algoritmą informacijos blokų generavimui. Vėliau, sukuriamas padalijimų medis, kuris susistemina šiuos informacijos blokus ir atitinkami atributai paskaičiuojami.

## 2.6. Padalijimų medžio generavimas

Panaudojus puslapio skaidymo algoritmą, gavome informacijos blokus [13]. Jie yra traktuojami kaip lapų mazgai padalijimo medžio struktūroje. Naudodami skyriklių paieškos algoritmą, sukursime kitas padalijimo medžio dalis. Pasitelkime pavyzdį, iliustruojantį šį algoritmą. Tarkime turime žiniatinklio puslapio keturis informacinius blokus, vardais: 1, 2, 3, 4. Jų išdėstymas pateiktas 6 pav. Pirmiausia yra surandamas horizontalus skyriklis, kuris originalų puslapį dalija į du blokus, su bloku numeriu 1 viršuje ir likusiu bloku – apačioje. Mes sugeneruojame vidinę viršūnę (node) padalijimų medžio viršuje šiam blokui ir pavadiname jį vardu:  $h$ , dėl to, kad jis horizontalus ( $h$ ) skyriklis. Kairysis šios viršūnės vaikas bus blokas 1, o šios viršūnės dešinysis vaikas – bus kita vidinė viršūnė, kuri bus gauta iš 2, 3, 4 bloko. Paieškos algoritmas yra rekursyvus ir vykdomas, kol visi informacijos blokai yra prijungti (prikabinti) ir sudaro medį. Galutinis padalijimų medis pateiktas 6b pav.

Jeigu žiniatinklio puslapio informacijos blokai yra iš dalies persidengiantys vieni su kitais, tokie kaip iššokantys paveikslukai ir meniu, paieškos algoritmas tinkamai nedirbs. Vis dėlto, pateikiamas variantas, kuriame nėra persidengiančių blokų. Kartais gali būti taip, kad gali būti daugiau nei vienas galimas skyriklis. Tuomet pasirenkamas vienas, kuris yra arčiausiai žiniatinklio puslapio viršaus.

Nors šis algoritmas yra rekursyvus, jo sudėtingumas yra tik linijinis. Jeigu bendras blokų skaičius yra  $N$ , tada dvejetainio padalijimo medžio dydis bus  $2N-1$ . Kiekviename skyriklių paieškos algoritmo žingsnyje, mes sugeneruosime naują vidinę medžio viršūnę. Apskaičiavimo sudėtingumas bus  $N-1$  (vidinių viršūnių skaičius).

### 2.6.1. Informacijos svarbumo reikšmė

Informacijos blokų svarbumas žiniatinklio puslapyje gali būti nustatomas pagal daugelį faktorių: turinį, jo dydį, padėtį. Pavyzdžiui, informacijos bloko vieta gali nurodyti jo svarbumą. Todėl daugelis žiniatinklio puslapių autorių paprastai svarbią informaciją talpina toje vietoje, kurią geriausiai mato vartotojas, tokiu principu vadovaujamosi ir atvaizduojant informacinius blokus mobilaus įtaiso ekrane[14].

Blokų reikšmingumo modelis yra apibrėžiamas kaip funkcija su požymiais ir charakterizuojama taip:

$$\langle \text{bloko požymiai} \rangle \rightarrow \text{bloko svarbumas} (2)$$

Kaip parodyta 2 lentelėje, svarbumas gali būti trijų lygių:

2 lentelė. Bloko svarbumo lygiai

3	Geriausiai matomos puslapio dalys, tokios kaip antraštė, pagrindinis tekstas ir t.t
2	Naudinga informacija, bet nelabai svarbi pateikti puslapio viršuje, tai nuorodos, su tema tiesiogiai nesusijęs turinys,
1	Kopijavimo teisių ženklai, dekoravimo elementai

Kiekvienam blokui apibrėžtas bruožų vektorius sudarytas iš 10 erdvinių ir 9 turinio požymių, tokių kaip pozicija, bloko dydis, teksto ilgis, nuorodų skaičius ir t.t.

### 2.6.2. Kiti atributai

Pritaikomumas (ADJ) nusako, ar informacijos bloke yra tokios HTML žymės, kaip <form>, <table>, <tr> arba <td>. Jeigu tokios žymos egzistuoja, tuomet informacijos blokui proporcingumo aspektas (aspect ratio) nebus taikomas. Tenka apibrėžti alternatyvas informacijos blokui. Tai galima gauti remiantis informacijos ištraukimo skaičiavimais. Čia nekalbėsiu apie teksto apibendrinimo algoritmą. Paminėsiu tik tiek, kad jeigu yra bet kuri žymė panaši į H1 arba H2 bloke, tuomet bus naudojamas šios žymos turinys.

## 2.7. DRESS pagrindu sukurtų dokumentų pritaikymas

Kalbėsime apie informacijos tikslumo koncepciją ir algoritmą, kuriuo galėsime susikurti optimaliaus žiniatinklio dokumento sluoksnius ir pritaikyti skirtingo dydžio mobilių įtaisų ekranams.

### 2.7.1. Informacijos tikslumas

Tai suvokimas “žiūrėk ir pajausk”. Informacijos tikslumo reikšmės kinta nuo 0 (žemiausias, visa informacija prarasta) iki 1 (aukščiausias, visa pateikiama informacija kaip originaliame puslapyje).

Tikslas - padidinti informacijos tikslumą, kuris yra pateikiamas galutiniam vartotojui mobiliaus įtaiso ekrane. Individualių pritaikytų blokų informacijos tikslumas sąlygojamas eilės parametrų, tokių kaip erdvinė ekrano sritis, turinio sumažinimas, spalvos gylis arba piešinio suspaudimo koeficientas.

Puslapiui  $P$ , sudarytam iš skirtingų blokų, galutinis informacijos tikslumas yra apibrėžiamas kaip visų blokų informacijos tikslumų suma. Mes galime panaudoti DRESS svarbumo reikšmę kaip puslapio kokybės svorį. Tokiu būdu, informacijos tikslumas galutiniame puslapyje bus [12]:

$$T = \sum_{i=1}^N w_i \cdot T_i$$

Buvo minėta, kad IF reikšmė priklauso nuo turinio blokų versijų, t.y. ar jis yra apibendrintas ar ne. Jeigu informacijos blokas turi alternatyvą, šio bloko IF reikšmė bus 0, priešingu atveju 1.

### 2.7.2. Problemos apibrėžimas

Čia  $P'$  - neapibendrintos informacijos blokų rinkinį žiniatinklio puslapyje:

$P' \subset P = \{B_1, B_2, \dots, B_N\}$ . Tikslas – rasti blokų rinkinį  $P'$ , kuris talpina informacijos tikslumo reikšmes. Laikome, kad visi turinio blokai turi vienodas galimybes būti naudojami, tokiu būdu apribojimai matematiškai užrašomi išraiška:



(a)

3 pav. Žiniatinklio puslapio pavyzdys (a)

Remiantis literatūra [16] galima suformuluoti šitokius apribojimus žiniatinklio puslapio sudalijimui:

#### 1 apribojimas

Informacinių blokų ir juose naudojamų simbolių suma neturi viršyti ekrano dydžio ir santraukos simbolių sumos skirtumo.

#### 2 apribojimas

Suformuotos santraukos ir joje naudojamų simbolių skirtumo suma neturi viršyti ekrano dydžio.

#### Tikslo funkcija

Naudojamų simbolių ir santraukos dydžių skirtumo suma neturi būti didesnė už ekrano dydžio ir santraukos simbolių sumos skirtumą.

Galime pastebėti, kad ši tinklo funkcija ekvivalenti supjaustymo problemai. Prieš tai esantis apribojimas kad ekrano dydis negali būti labai mažas; kitaip, mes negalėsime surasti efektyvaus

sluoksnio netgi tuomet, kai visi blokai buvo apibendrinti. Šiuo retu atveju, sub-medžio apibendrinimas taps būtinas.

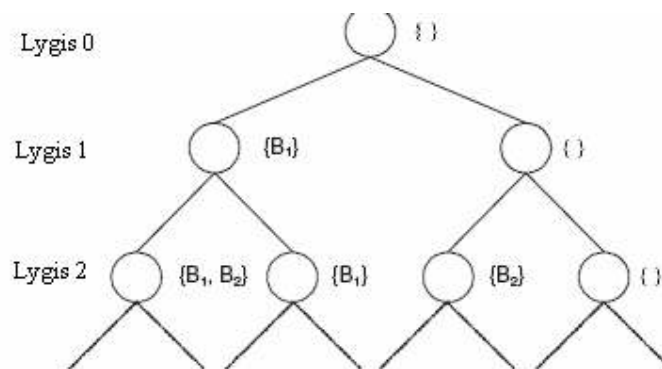
1 apribojimas lygtis negarantuoja, kad MPH ir MPW bus patenkintos. Siūloma problemą spręsti dviem lygiais: Pirmia, naudoti šakų ir ribų algoritmą, kad išskaičiuotume visus galimus blokų  $P'$  rinkinius, tada kiekvienam bloko rinkiniui, naudodami talpumo koeficiento padalijimo algoritmą, kad nustatytume, ar efektyvus sluoksnis gali būti surastas. Proceso metu, mes ieškosime bloko tarp galimų blokų rinkinio  $P'$ , išsirinkdami vieną optimaliausią, t.y. didžiausios IF reikšmės.

### 2.7.3. Puslapio atvaizdavimo (rendering) algoritmas

Kalbant apie NP problemą, yra pasiūlytas patobulintas šakų-ribų algoritmas, kuris leidžia išsirinkti efektyvesnę bloką.

### 2.7.4. Bloko rinkinio pasirinkimas

Kaip parodyta 3 paveiksle, suformuojame dvejetainį medį blokų išrinkimui, kur aukščiausia viršūnė yra nulinė, tai reiškia, kad visi blokai yra apibendrinti ir mes turime:



5 pav. Dvejetainis medis gautas pasirenkant optimalių blokų rinkinį

- Kiekviena viršūnė reiškia neapibendrintų blokų rinkinį
- Kiekvienas lygis reiškia informacijos blokų įtraukimą
- Kiekvienas išsišakojimas reiškia pasirinkimą imant originalų turinį arba sukuriant bloko santrauką kitame lygyje [16].

Tokiu būdu, šio medžio aukštis yra  $N$ , bendras blokų skaičius žiniatinklio puslapio viduje ir kiekviena lapa viršujė šiame medyje atitinka skirtingas bloko rinkinio  $P'$  galimybes. Kiekvienai dvejetainio bloko medžio viršūnei yra ribota IF reikšmė, ji gali būti įgyjama einant tarp visų jos sub-medžių. Akivaizdu, kad žemiausia riba bus tada, kai nėra nei vieno nepažymėto bloko.

Igyvendiname skverbimąsi medžiu gilyn pagal šiuos apribojimus:

- Bet kada aukštesnė viršūnės riba mažesnė nei geriausia IF reikšmė šiuo metu įgyta. Šios viršūnės sub-medžio visuma bus nupjauta.
- Tuo pačiu metu, kiekvienai viršūnei mes tikriname srities dydžio apribojimus, pagal (4) lygtį, kad patikrinti jos pagrįstumą.
- Jeigu mes pasiekiamo blokų rinkinį, kurio IF reikšmė didesnė negu dabartinė geriausia IF reikšmė, turime tikrinti šio sprendinio įvykdomumą, naudodami sluoksnių algoritmą, aprašomą skyriuje 5.3.2. Jeigu sprendinys įvykdomas, geriausią IF reikšmę pakeičiame pagal vieną jų.

Tikrindami galimų IF reikšmių ribas ir kiekvieno bloko rinkinio sluoksnių pagrįstumą, skaičiavimo kaštai yra žymiai sumažinami. Taip pat galima panaudoti kitą techniką, leidžiančią sumažinti laiką, tokią kaip visų blokų sutvarkymą didėjančia tvarka pagal jų svarbumo reikšmes bei pradėti jų paiešką. Keletas žiniatinklio puslapis talpina didelį informacijos blokų skaičių, taigi, bet koks šakų-ribų algoritmas bus lėtas. Šiuo atveju, naudojamas godus algoritmas, kuriuo generuojamas sub-optimalus rezultatas vietoj to, kad bandyti gauti optimalų sprendimą. Godus algoritmu paprastai bandoma įkelti informacijos bloką į blokų rinkinį, po vieną didinant jų svarbumo indeksą. Jeigu blokas negali būti įkeltas į rinkinį, jis toliau nebus vertinamas. Šio būdo efektyvumas buvo įrodytas sprendžiant kuprinės problemą. Jeigu blokų skaičius didesnis nei 15, godus algoritmas bus naudojamas skaičiavimų pagreitinimui.

### 2.7.5. Talpos padalijimas

Vykdėme blokų rinkinio  $P'$  formavimą, bandėme surasti sluoksnį, į kurį sudėtume visus blokus ir apibendrintume. Pagrindinė žiniatinklio sluoksnių optimizavimo problema yra panaši į aukštų planavimą, kuri žinoma, kaip NP problema. Panaudosime talpos padalijimo metodą, kuriuo problemą spęsimė dviem žingsniais (pavaizduota 4 pav):

Pirma, mes eisime per padalijimų medį naudodami „žemyn-aukštyn“ metodą, skaičiuodami talpos, aukščio, pločio apribojimus kiekvienai viršūnei. Paskui padalijimų medis skaičiuojamas „aukštyn-žemyn“ metodu.

Procedūra detaliau pateikiama 5 pav., kur yra apskaičiuota talpa, mažiausias aukštis ir plotis kiekvienai viršūnei.

Antra, palyginame dviejų viršūnių sub-medžių vidinių viršūnių talpas ir žiūrime, kad padalijimų skaičius šios viršūnės būtų proporcingas joms. 6 pav. pateikiamas algoritmas detaliau.

Šio algoritmo sudėtingumas yra  $O(N)$ , abu žingsniai gali būti įgyvendinami per  $O(N)$  laiko. Talpos algoritmas yra pakankamai greitas tikrinti kiekvieno pažymėto bloko sluoksnių įvykdomumą. Pavyzdys pateikiamas 11 pav. MPS keturių turinio blokų puslapyje yra 1000, 1000, 2500 ir 2500. Sakykime, kad visi jie nėra apibendrinti pritaikymui, aukščiausios viršūnės padalijimų skaičius bus:  $100/(1000+2500+2500)=0.14$ . Jeigu ekrano dydis 600 aukščio taškų, mes turėsime suskaldyti

horizontalią sritį į eilutes:  $600 \times 0.14 = 84$ . Ši procedūra atliekama rekursyviai, iki tol, kol visi padalijimų skaičiai yra aiškūs (Algoritmų kodai paimti iš originalaus šaltinio [16]).

```
Slicing ()
{
    // calculate capacity, height and width constraints
    Aggregate (root);

    // calculate slicing numbers
    root->area = TARGET_AREA; // target display size
    root->height = TARGET_HEIGHT; // target display height
    root->width = TARGET_WIDTH; // target display width
    if (Allocate (root) == true) return true;
    else return false;
}
```

6 pav. Talpos padalijimų algoritmas

```
Aggregate (p )
{
    if p is a leaf node
        if p is a summarized block
            p->capacity = size (p->alternative);
            p->minheight = height (p->alternative);
            p->minwidth = width (p->alternative);
        else // p is unsummarized
            p->capacity = p->MPS;
            p->minheight = p->MPH;
            p->minwidth = p->MPW;
    else // p is an inner node
        Aggregate (p->lchild );
        Aggregate ( p->rchild );
        p->capacity = p->lchild->capacity + p->rchild->capacity;
        if p->label == vertical
            p-> minheight = max(p->lchild->minheight, p->rchild->minheight );
            p-> minwidth = p->lchild->minwidth + p->rchild->minwidth;
        else // p's label is horizontal
            p-> minheight = p->lchild->minheight + p->rchild->minheight;
            p-> minwidth = max (p->lchild->minwidth, p->rchild->minwidth);
}
```

7 pav. Algoritmas skaičiuojantis talpos, aukščio, pločio apribojimus kiekvienai viršūnei

```

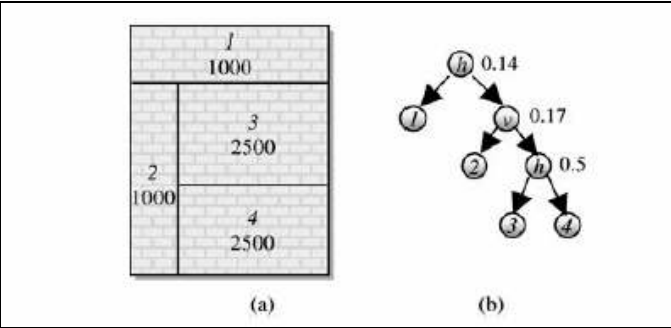
Allocate (p)
{
  if p->area < p->capacity or p->height < p->minheight
    or p->width < p->minwidth
    return false;

  if p is an inner node
    p->slicingnumber = p->lchild->capacity / p->capacity;
    if p->label == vertical
      p->lchild->height = p->height
      p->rchild->height = p->height;
      p->lchild->width = p->slicingnumber * p->width;
      p->rchild->width = p->width - p->lchild->width;
      if p->lchild->width < p->lchild->minwidth
        p->lchild->width = p->lchild->minwidth;
        p->rchild->width = p->width - p->lchild->minwidth;
      else if p->rchild->width < p->rchild->minwidth
        p->rchild->width = p->rchild->minwidth;
        p->lchild->width = p->width - p->rchild->minwidth;
    else // p's label is horizontal
      p->lchild->width = p->width
      p->rchild->width = p->width;
      p->lchild->height = p->slicingnumber * p->height;
      p->rchild->height = p->height - p->lchild->height;
      if p->lchild->height < p->lchild->minheight
        p->lchild->height = p->lchild->minheight;
        p->rchild->height = p->height - p->lchild->minheight;
      else if p->rchild->height < p->rchild->minheight
        p->rchild->height = p->rchild->minheight;
        p->lchild->height = p->height - p->rchild->minheight;

  if (Allocate (p->lchild ) == false) return false;
  else if (Allocate (p->rchild ) == false) return false;
  else return true;
else return true // do nothing when p is a leaf node
}

```

8 pav. Algoritmas, suformuojantis galutinį ekrano stačiakampį kiekvienai viršūnei [16]



9 pav. Žiniatinklio sluoksnis su MPS etikete (a) ir padalijimų medis su padalijimų skaičių etiketėmis (b) [16]

2.7.6. Turinio pritaikymas

Po sluoksnių optimizavimo, kiekvienam blokui, ar apibendrintas ar ne, būna priskirta stačiakampio formos vieta ekrane. Pagal padalijimo algoritmą, paskirtos vietos dydis kiekvienai viršūnei bus didesnė už jos MPS reikšmę. Aukštis ir plotis ekrane bus ribojami apribojimų kiekvienam blokui. Jeigu blokas yra nurodytas kaip pritaikomas, mes jį priderinsime prie stačiakampio formos didindami arba mažindami. Tai vykdoma, kad būtų išlaikytas proporcingumas.

## 2.8. DRESS metodo realizacija mobiliuose įtaisuose

### 2.8.1. DRESS realizacija naršyklėse

Prototipe buvo realizuotas DRESS metodas.

```
<!-- DRESS SlicingTree = ( ( A - ( B | ( C | ( D - E ) ) ) - F ) -->
<!--Block ID="A" IMP="0.15" MPS="35000" MPA="80" MPW="400"
      ADJ="No" ALT="Google News Search" -->
contents of block A ...
<!--Block ID="B" IMP="0.1" MPS="14000" MPA="140" MPW="100"
      ADJ="No" ALT="Left Sidebar" -->
contents of block B ...
<!--Block ID="C" IMP="0.4" MPS="80000" MPA="150" MPW="200"
      ADJ="Yes" ALT="Top Stories" -->
contents of block C ...
<!--Block ID="D" IMP="0.2" MPS="30000" MPA="100" MPW="150"
      ADJ="Yes" ALT="Hot News List" -->
contents of block D ...
<!--Block ID="E" IMP="0.1" MPS="20000" MPA="100" MPW="200"
      ADJ="No" ALT="In the news" -->
contents of block E ...
<!--Block ID="F" IMP="0.05" MPS="9000" MPA="40" MPW="150"
      ADJ="Yes" ALT="Google Footer" -->
contents of block F ...
<!-- DRESS End -->
```

10 pav. DRESS metodo kodas HTML faile [16]

naršyklė gali automatiškai parinkti ir apibendrinti būtinas žiniatinklio puslapio dalis, tada optimizuoti reikiamos srities sluoksnius.

Čia nebuvo bandoma pateikti naujo žiniatinklio puslapio formato, bet buvo naudojamas puslapio HTML formatas. Prototipe DRESS struktūra saugoma kaip komentaras HTML faile arba stilių lentelėje. Failo viršuje užkoduota puslapio struktūra: “((A - (B|(C|(D - E)))) - F)”, kur “-” ir “|” reiškia horizontalius ir vertikalinius padalijimus. Kiti DRESS atributai yra parodyti 12 pav.

Kada peržiūrimas žiniatinklio puslapis, pirmiausia gramatiškai nagrinėjama viduje esanti DRESS informacija ir tada generuojamas pagrindinis HTML failas. Tuo pat metu blokų turinys išsaugomas laikiname HTML faile, kuris pasiekiamas alternatyvų nuorodomis. Tada tarpiniai duomenys gali būti išsaugomi atmintyje vietoj failo, jeigu adaptacija yra įvykdyta kliento pusėje. Jeigu DRESS informacija nėra randama žiniatinklio puslapyje, algoritmas pavaizduotas pav.4 generuoja DRESS automatiškai.

DRESS nereikalauja papildomos modifikacijos kliento įtaisuose. Transformacija gali būti įvykdyta turinio serveryje arba tarpiniuose proxy. Pavyzdžiui, DRESS proxy gali aptikti, ar web puslapis yra DRESS formate ir tada transformuoti jį į ne DRESS puslapį pagal vartotojo įtaiso ekrano dydį.

### 2.8.2. Algoritmo patikrinimas

Darbe pateikiamas užsienio tyrinėtojų atliktas tyrimas ir gauti rezultatai, nes man nebuvo techninių galimybių išbandyti vykdomą transformavimo procesą, nes tam reikia delninio mobilaus įtaiso bei



specialios serverinės programinės įrangos, galinčios vykdyti DRESS metodo komentarų aptikimo ir generavimo procesus.

Vykdyto tyrimo aprašymas:

Dalyvavo 6 žmonės, kurie buvo susipažinę su žiniatinklio puslapių peržiūra ir mokėjo naudotis mobiliais įtaisais. Kiekvienas vartotojas turėjo įvykdyti 6 žiniatinklyje paskelbtas užduotis atitinkama tvarka.

Vartotojai buvo suskirstyti į dvi grupes. Viena grupė pirmiausia naudojo DRESS metodu sugeneruotus puslapius, o paskui thumbnail metodu suprojektuotus puslapius. Kita grupė, atvirkščiai – pirmiau naudojo thumbnail, o paskui DRESS. Buvo peržiūrimi žiniatinklio puslapiai. Ekranų dydis 240x320.

Visi aplankyti puslapiai ir atitinkamos trukmės buvo įrašomos ir naudojamos vėlesniai analizei.

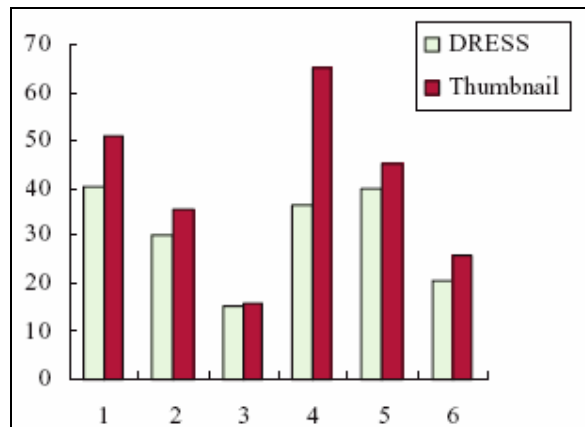
Vidutinis kiekvieno vartotojo peržiūrėjimo laikas pateiktas 14 pav. Kaip ir buvo tikėtasi, DRESS metodu sugeneruotuose puslapiuose informacija buvo pasiekama daug greičiau nei thumbnail metodu suformuotuose. Pagal vidurkį, DRESS, peržiūrėjimo laikas sumažėja 23.5%.

Vartotojai sakė, kad naudodamiesi DRESS, jie gali skaityti tekstą, kol thumbnail metodas vykdo puslapio transformavimą į mažą paveiksluką. Nors automatiškai sugeneruoti DRESS apibendrinimai nėra dar pilnai išbaigti, tačiau šis metodas daug efektyvesnis.

Eksperimentas taip pat parodė algoritmų efektyvumą, įvertinant skaičiavimo kaštus. Teste naudota techninė įranga:

Dell Optiplex GX 270 using a 2.8 GHz Pentium

4 processor, 512 MB of RAM, and Windows XP Professional system.



11 pav. Vidutinis naršymo laikas skirtinguose sąsajose

Rezultatas - DRESS generavimo kaštų vidurkis 27.5 milisekundės, o DRESS atvaizdavimo kaštų vidurkis 35.2 milisekundės. Rezultatas rodo, kad be kodo optimizavimo, algoritmas yra pakankamai greitas naudoti realiose žiniatinklio sistemose bet kur – žiniatinklio serveriuose ar proxy serveriuose, arba netgi vartotojo mobiliame įtaise [16].

### III dalis – magistrinio darbo projekto dalies transformavimo modelis

#### Matematinis modelis

Čia  $P'$  - neapibendrintos informacijos blokų rinkinį žiniatinklio puslapyje:  $P' \subset P = \{B_1, B_2, \dots, B_N\}$ .

Tikslas – rasti blokų rinkinį  $P'$ , kuris talpina informacijos tikslumo reikšmes. Laikome, kad visi turinio blokai turi vienodas galimybes būti naudojami, tokiu būdu apribojimai matematiškai užrašomi išraiška:

$$\sum_{B_i \in P'} \text{dydis}(ALT_i) + \sum_{B_i \in P'} MPS_i \leq \text{Sritis} \quad (4)$$

kur Sritis – srities dydis, dydis x – funkcija, kuri grąžina ekrano srities dydį, reikalingą ALTi naudojimui. Jeigu atsižvelgiame į mobilaus įtaiso ekrano apribojimus, transformacija vykdoma pagal išraišką:

$$\sum_{B_i \in P'} (MPS_i - \text{dydis}(ALT_i)) \leq \text{Sritis} - \sum_{B_i \in P} \text{dydis}(ALT_i) \quad (5)$$

tuomet žiniatinklio puslapio sluoksnių optimizavimo problema tampa:

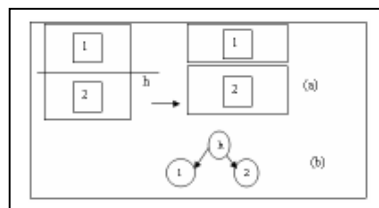
$$\begin{aligned} \max \left( \sum_{B_i \in P'} IMP_i - IF_{B_i} \right) &= \max_{P'} \left( \sum_{B_i \in P'} IMP_i \right) \\ \sum_{B_i \in P'} (MPS_i - \text{dydis}(ALT_i)) &\leq \text{Sritis} - \sum_{B_i \in P} \text{dydis}(ALT_i) \end{aligned} \quad (6)$$

[16]

Galime pastebėti, kad ši lygtis yra ekvivalenti tradicinei 0-1 kuprinės (knapsack) problemai. Prieš tai esanti lygtis reiškia, kad ekrano dydis negali būti labai mažas; kitaip, mes negalėsime surasti efektyvaus sluoksnio netgi tuomet, kai visi blokai buvo apibendrinti. Šiuo retu atveju, sub-medžio apibendrinimas taps būtinas. (4) lygtis negarantuoja, kad MPH ir MPW bus patenkintos. Siūloma problemą spręsti dviem lygiais: Pirma, naudoti šakų ir ribų algoritmą, kad išskaičiuotume visus galimus blokų  $P'$  rinkinius, tada kiekvienam bloko rinkiniui, naudosis talpumo koeficiento padalijimo algoritmą, kad nustatytume, ar efektyvus sluoksnis gali būti surastas. Proceso metu, mes ieškosime bloko tarp galimų blokų rinkinio  $P'$ , išsirinkdami vieną optimaliausią, t.y. didžiausios IF reikšmės.

Modelio veikimą pademonstruosiu pavyzdžiais, kurie buvo praktiškai realizuoti):

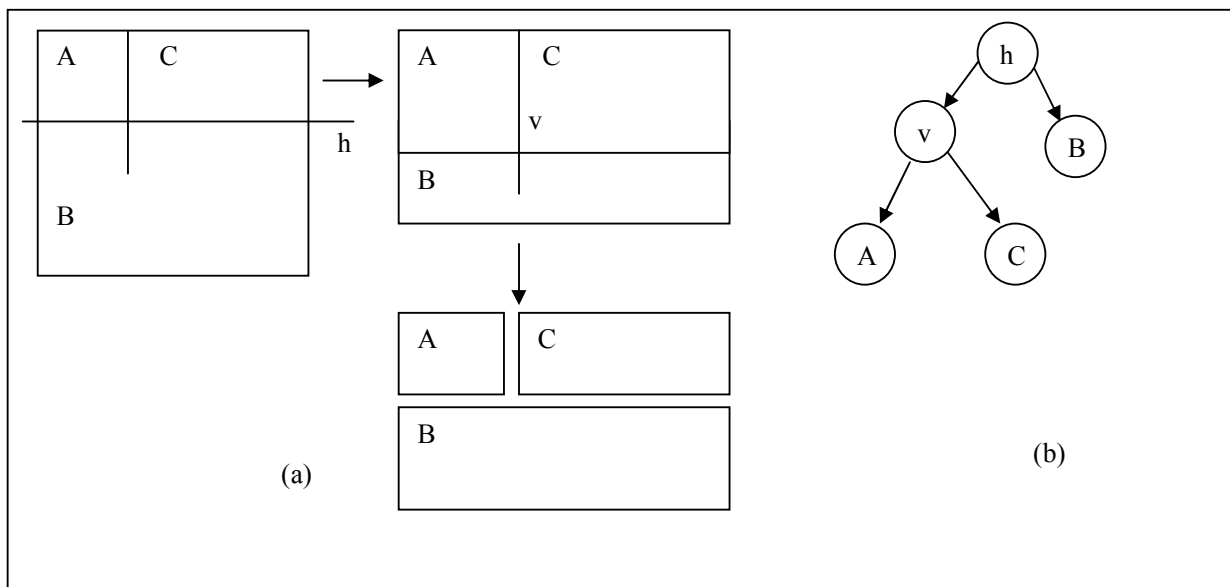
#### 1 skaidomas puslapis – skelbimų talpinimo forma



12 pav. Skelbimų talpinimo forma 13 pav. Skyriklių (separator) paieškos algoritmo taikymas (a) ir galutinis padalijimo medis (b)

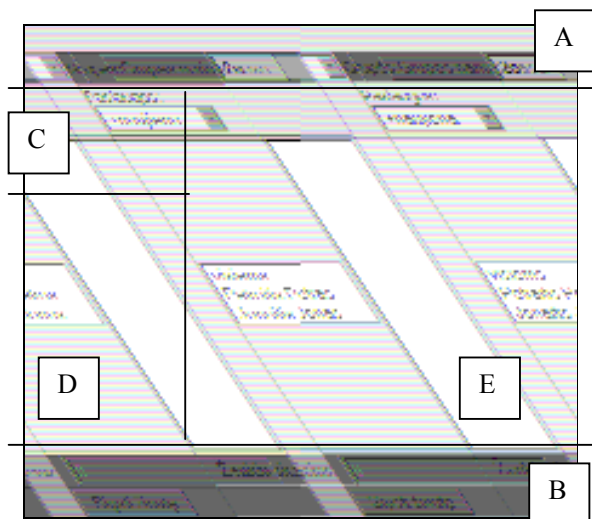
#### 2 skaidomas puslapis- elektroninis dienynas

14 pav. Elektroninio dienyno puslapis

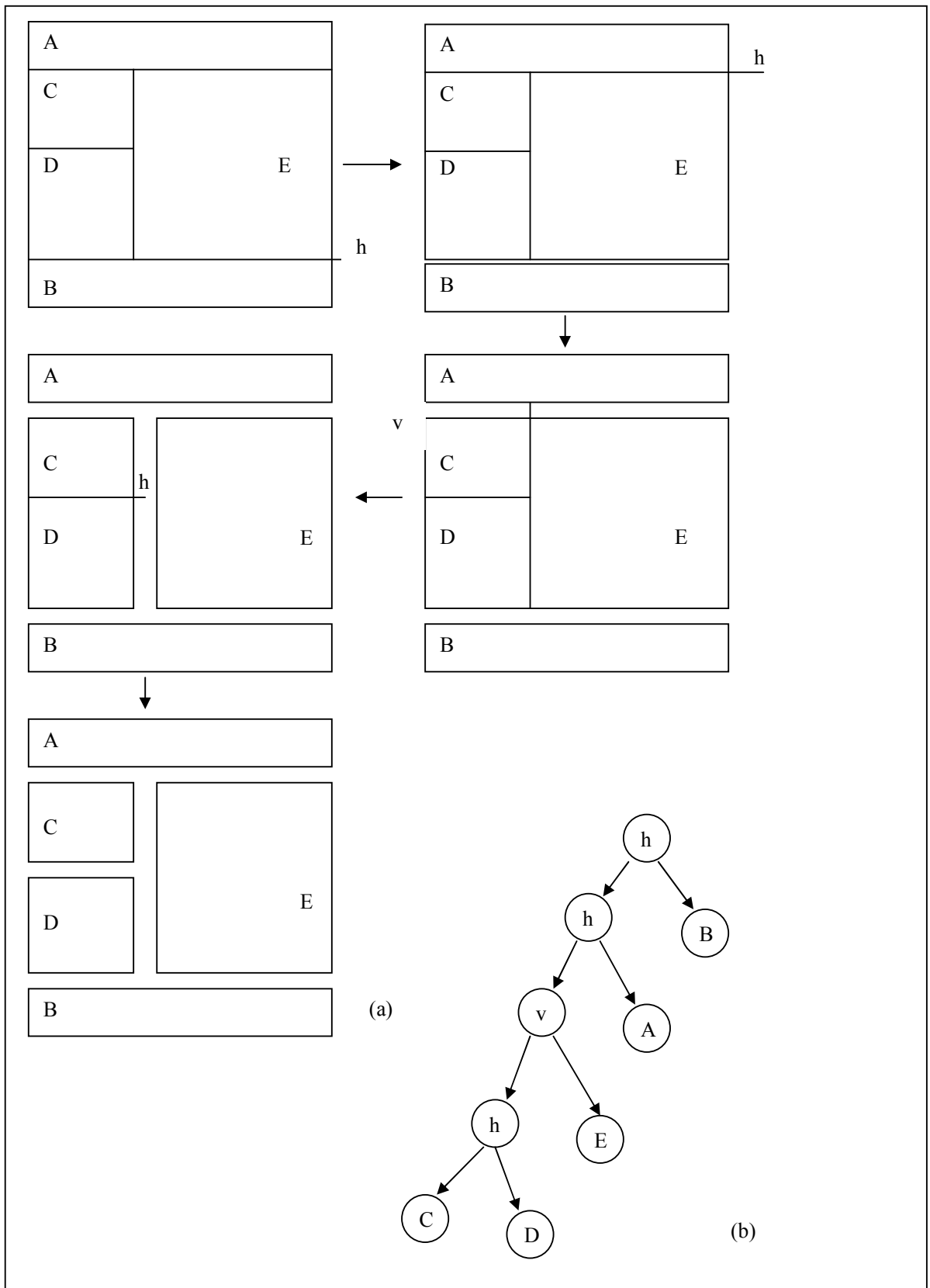


padalijimo medis (b)

3 skaidomas puslapis – SMS siuntimas kompiuteriu



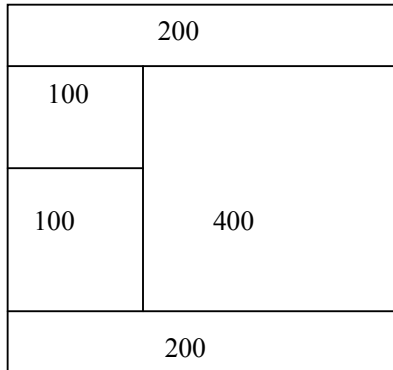
16 pav. SMS siuntimo kompiuteriu forma



17 pav. El. dienyno puslapio skaidymas (a) ir galutinis padalijimo medis (b)

*Informacinių blokų dydžio formavimas*

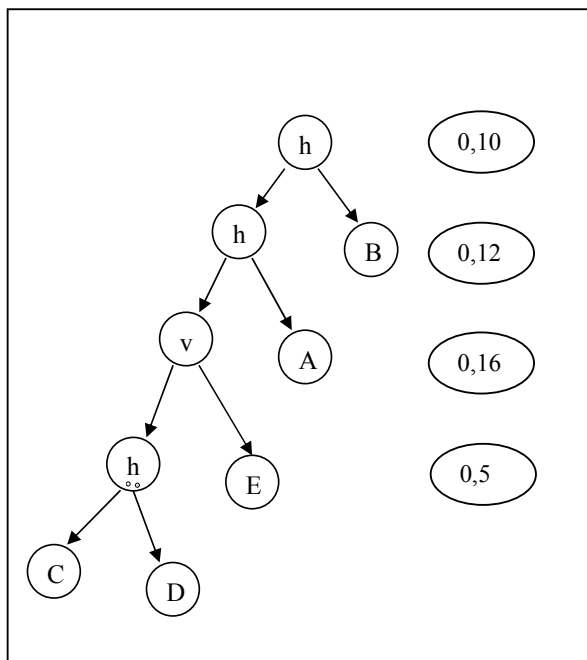
Ekranas dydis yra žinomas. Tuo remiantis galima formuoti informacijos blokus, kuriuos bus galima atvaizduoti ekrane. Tarkime ekrano dydis 500 x 600 taškų. Turime 24 pav. puslapio struktūrą:



18 pav. SMS siuntimo puslapio struktūra

Tarkime, kad kiekvienas blokas užima atitinkamą ekrano taškų skaičių: 1000, 1000, 700, 700 ir 2500. Sakykime, kad visi informacijos blokai nėra apibendrinti pritaikymui, tokiu būdu aukščiausios viršūnės padalijimų skaičius bus:  $100/(200+200+100+100+400)=0,10$ . Jeigu ekrano aukštis 600 taškų, mes turėsime suskaldyti horizontalią sritį į eilutes:  $600 \times 0,14 = 60$ . Ši procedūra atliekama rekursyviai, iki tol, kol visi padalijimų skaičiai yra aiškūs.

Padalijimų medis su padalijimų skaičių reikšmėmis:



19 pav. Padalijimų medis su padalijimų skaičių reikšmėmis

DRESS kodas, kuris bus sugeneruotas ir patalpintas .html faile atrodys taip

```
<!--DRESS padalijimu medis = ((B|(E-C-D)-(A))-->
<!--Block id="A" IMP="0.15" MPS="50000" MPH="80"
MPW="400" ADJ="No" ALT="grupes_pasirinkimas"-->
<!--Block id="B" IMP="0.1" MPS="50000" MPH="60"
MPW="300" ADJ="No" ALT="simboliu_kiekis"-->
<!--Block id="C" IMP="0.15" MPS="50000" MPH="80"
MPW="400" ADJ="No" ALT="paslaugos"-->
<!--Block id="D" IMP="0.20" MPS="50000" MPH="80"
MPW="400" ADJ="No" ALT="pavardes"-->
<!--Block id="E" IMP="0.40" MPS="50000" MPH="80"
MPW="600" ADJ="No" ALT="sms_tekstas"-->
<!--DRESS End-->
```

20 pav. Galimas DRESS kodas .html faile pagal aukščiau pateiktą variantą

Darbo metu buvo atliktas rankinis informacijos transformavimas ir pritaikymas prie atitinkamo dydžio mobilaus įtaiso ekrano. Darbo metu susipažinta su transformavimo procesu, parametrais, vykdymo tvarka. Praktiškam realizavimui reikia suformuotą DRESS kodą patalpinti į serverį bei turėti nešiojamą delninį kompiuterį.

### 3. PROJEK TINĖ DALIS

Magistrinio projekto realizavimo metu apibrėžtas sistemos tikslas – sukurti mob. mokymosi modulį, leidžiantį informaciją skaityti ir ją siųsti į/iš kuriamą el.mokymosi portalą mobiliu telefonu arba kompiuteriu. Prisijungimui bus naudojami mobilūs telefonai, palaikantys sms ir wap technologijas. Mob. mokymosi moduliui naudosis KTU gimnazijos nariai – administracija, mokytojai, mokiniai, tėvai (globėjai).

Sistemos projektui sudaryti pasirinktas objektinis projektavimo metodas. Projektavimui bus naudojama Microsoft Project vaizdinio modeliavimo kalba, kuri leidžia pakankamai tiksliai ir vaizdžiai atvaizduoti kuriamą modulį.

Sistemos architektūra pateikiama keliais aspektais: panaudos atvejų (PA), procesų, išdėstymo ir realizavimo. Visa tai pateikiama diagramose. Sistemos architektūra pateikiama remiantis UML rekomendacijomis. Sistemos specifikacija pateikiama:

- Panaudos atvejų vaizdu (panaudojimo atvejų diagrama UML diagrama);
- Loginis programos vaizdas (Klasių diagrama, skaidymas į paketus);
- Procesų vaizdas (sekų diagramos);

#### 3.1. Architektūros tikslai ir apribojimai

Architektūrinius sprendimus įtakojantys reikalavimai:

- Sistema turi būti realizuota kaip kliento – serverio sistema.
- Sistemos funkcijos turi būti prieinamos naršykle bei mobiliu telefonu.
- Sistema turi užtikrinti kaupiamų duomenų saugumą. Sistemos vartotojui prieinami duomenys turi priklausyti nuo jo teisių .
- Sudarant sistemos architektūrą, turi būti atsižvelgta į būtinas programos vykdymo charakteristikas, apibrėžtas reikalavimų specifikacijoje.
- Projektavimo įrankiai: Rational Rose, MS Word.

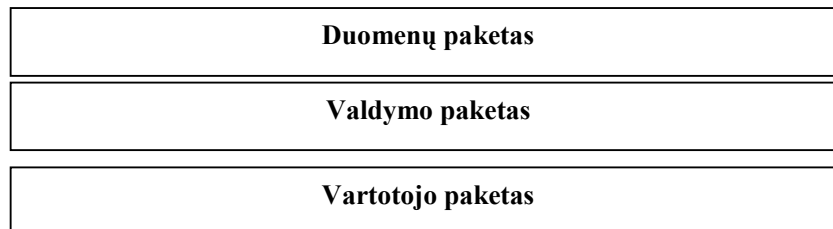
M.mokymosi sistemos valdymo loginis modelis sudarytas pagal daugiasluoksnės architektūros modelį.

Daugiasluoksnės architektūros teikiami privalumai:

1. Aiškesnis sistemos programinis kodas, lengvesnis sistemos palaikymas. Kadangi kodas, atliekantis panašias funkcijas, yra laikomas vienoje vietoje (o nėra išmėtytas po visą sistemą), tai jį tampa lengviau prižiūrėti bei koreguoti;

2. Darbų pasidalinimas ir sistemos kūrimo darbo išlygiagretinimas. Galima išskirti tris programuotojo roles – duomenų bazių projektavimas, taikomosios srities taisyklių įgyvendinimas bei vartotojų sąsajos kūrimas.

Naudotas trijų lygių architektūros modelis:



21 pav. Trijų lygių architektūros modelis

Kiekvienas paketas atitinka veiklos tipą:

- Duomenų paketas – duomenų struktūrų projektavimas.
- Valdymo paketas – programos (sistemos funkcionalumo) projektavimas. Veiklos paketą sudaro valdančios klasės, kuriomis įgyvendinamos veiklos taisyklės
- Vartotojo paketas, kurį sudaro formos (ribinės klasės), kurių pagalba sistemos vartotojai bendrauja su sistema. Šiame sluoksnyje yra prisijungimo forma, sistemos administravimo formos.

Sudaryta sistemos architektūra tenkina pagrindinius kokybės reikalavimus:

1. Sistemos funkcijos bus prieinamos per mobilaus telefono naršyklę.
2. Vartotojo sąsajos formavimui naudojami tik standartiniai naršyklės palaikomi valdymo elementai ir iš jų pagaminti komponentai.
3. Sistema užtikrins sankcionuotą duomenų priejimą.
4. Informacinė sistema bus funkcionali 24 valandas per parą, 7 dienas per savaitę.
5. Visi sistemos vartotojai naudosis tik vėliausia sistemos versija, kuri bus įdiegta serveryje.
6. Automatizuotas duomenų bazes kopijų darymas, užtikrins duomenų išsaugojimą, net įvykus techniniam serverio gedimui

Veiklos funkcijų realizavimas ir laikymas vienoje vietoje įgalina lengvai atlikti pakeitimus.

Sistema yra lengvai pernešama į kitą vietą.



### 3.1.1. Apribojimai sprendimui

Sistema turi būti prieinama bet kuriuo paros metu, naudojant bet kurių mobilių telefonų kompanijos telefoną, palaikantį sms ir wap technologijas.

Pagrindiniai apribojimai šio tipo moduliui:

*laikas* - modulis turi veikti greitai; ji turi pakankamai greitai persiųsti žinutes jų adresatams – taip nariai greitai galės reaguoti į iškilusius klausimus;

*naudojimo paprastumas* – nario neturi apsunkinti darbo aplinka, bei konkrečiu momentu nereikalingos funkcijos;

ataskaitų informatyvumas, aiškumas, paprastumas.

### 3.1.2. Diegimo aplinka

Serveris kuriame bus patalpintas sukurtas modulis turi:

- Palaikyti mobilų internetą, kurio įėjimo bei išėjimo greitis ne mažesnis nei 51 kbit/s.
- Palaikyti php ir wml formatus.
- Būti ne mažiau 1GHz greičio procesorius.
- 256MB darbinė atmintis.
- Kietas diskas nemažesnis nei 10 GB.

### 3.1.3. Bendradarbiaujančios sistemos

Sistema bendradarbiaus su pasirinktomis sistemomis:

- WEB serveris – Apache 2.0
- Operacine sistema – Windows.
- Preprocesorius – PHP.
- Duomenų bazė –MySQL 5.0.1
- Ozeki SMS server.

### 3.1.4. Komerciniai specializuoti programų paketai

Ozeki SMS server modulis bus integruotas į kuriamą modulį. Tai modulio biudžetą labai neįtakoja.

### 3.1.5. Numatoma darbo vietos aplinka

Sistema turi būti pasiekama wap ir sms žinute. Mobilaus modulio administratoriaus kai kurias funkcijas (duomenų valdymą) galės atlikti mobilaus interneto pagalba, kitas funkcijas – dirbdamas prie kompiuterio. Nes jis bus atsakingas už sms žinučių korektišką išsiuntimą, modulio programavimo darbus, vartotojų kūrimus ir t.t. Prie sistemos jis galės jungtis iš bet kurios vietos, kur yra kompiuterinis ir mobilus internetas. Kitų vartotojų – mokytojų – darbo vieta, jei ji susijusi su medžiagos talpinimu – bus darbas su kompiuteriu, kitais atvejais – darbus galima atlikti mobiliu telefonu. Mokiniai, tėvai svarbiausią informaciją gaus SMS žinute (tėvų susirinkimai, mokyklos naujienos) ir valdys mobiliu telefonu arba kompiuteriu (savo vaikų pažymių peržiūra, ataskaitų generavimas ir t.t). Informacija pateikiama internetu ir telefonu bus saugoma tame pačiame el.mokymo portale, o m-mokymo modulis įgalins jame esančią informaciją pasiekti ir valdyti mobiliu telefonu, naudojant sms ir wap technologijas.

### 3.2. Praktikos metu taikytos testavimo strategija:

- Vienetų testavimas
- Integravimo testavimas
- Priėmimo testavimas
- Aukšto lygio testavimas

#### 3.2.1. Vienetų testavimas

Testuota atskiri sistemos gabalai, naudojant baltos ir juodos dėžės principus. Sukurti funkcinį testų atvejai, jie papildyti paruoštais duomenimis, kurie patikrinti vykdant visus testuojamos dalies kelius. Tokiu būdu patikrinta ar visi moduliai dirba taip, kai turi dirbti.

#### 3.2.2. Integravimo testavimas

Šis testavimas atliktas, kai žinojau, kad atskiros sistemos dalys dirba teisingai. Testavime naudoju vieną iš integravimo testavimo metodų - smulkinantį testavimą. Susikūriau kamščius, kurie leido ištestuoti norimą sistemos lygį.

#### 3.2.3. Priėmimo testavimas

Šią testavimo strategiją įgyvendinau pas užsakovą, tam kad užtikrinti ar sistema dirba taip kaip pageidavo užsakovas. Tuo tikslu atlikau keletą juodos dėžės testavimų, t.y. buvo paruoši testo atvejai, su įėjimais ir laukiamais išėjimais. Aptiktos klaidas fisuotos ir šalintos.

### 3.2.4. Aukšto lygio testavimas

Galutinį testavimą atlikau pas užsakovą, kur vykdžiau šiuos skirtingo tipo testavimus:

**3.2..4.1. Stresinis testavimas** - tikrinau, kaip sistema veikia esant didesniai apkrovimui. Prisijungta kelių vartotojų vienu metu.

**3.2..4.2. Saugumo testavimas** – testuota, stebint ar prie reikiamų duomenų gali prisijungti tik tie vartotojai, kuriems priėjimas prie jų leidžiamas. Sistema neleido vartotojams trinti duomenų, kurių jiems nebuvo apibrėžta pagal specifikaciją.

#### 3.2.4.3. Testavimo resursai

Testavimui naudoju savo ir gimnazijos kompiuterio ir programinės įrangos resursus. Jų sąrašas:

Personalinis kompiuteris (AMD III, 512 MB RAM, 80 GB kietasis diskas).

PHP kompiliatorius

WAP debuggeris

#### 3.2..4.4. Testavimo įrankiai ir aplinka

Automatizuoto testavimo įrankiai pagrindinai yra mokami, tad juos kūriau pats. Testavimo aplinka - Windows aplinka su įdiegta PHP, MySQL duomenų baze, Apache Server, WAP kompiliatorius.

Testavimo rezultatai kaupiami tekstinio dokumento lentelėje, kurioje nurodytas testo numeris, jo aprašas, kas padaryta ir kokie gauti rezultatai.

## 4. TYRIMO DALIS

1-3 semestruose buvo sukurta programinė įranga, kuria galima kurti m.mokymosi modulius. Sukurta PĮ – tai eksperimentinis variantas ir jis atitiko visus reikalavimų specifikacijoje apibrėžtus reikalavimus. Tačiau rūpinantis tolimesnius sistemos kokybės tobulinimu, vykdomi šių komponentų, didinančių sistemos kokybę, tobulinimo darbai:

- Svarbios informacijos kodavimas patikimesniu md5 algoritmu
- Greitas rezultatų pateikimas pagal paieškos raktus
- Perėjimų tarp puslapių efektyvinimas.
- WML el.pašto kodas.

### 4.1. Svarbios informacijos kodavimas md5 algoritmu

Šiais laikais informacijos tinklai vis labiau stengiamasi apsaugoti, tačiau nėra visiškai saugūs, tad tenka paties programinės įrangos kūrėjas kurti arba taikyti kitų sukurtas funkcijas, kurios garantuoja siunčiamos informacijos patikimą užkodavimą vartotojui siunčiant informaciją bei atkodavimą kitam vartotojui gaunant pateikiamą informaciją. Sistemoje imtas taikyti daug patikimesnis md5 algoritmas

Algoritmo taikymas – md5 funkcijos kvietimas:

```
<?php
// Nurodote koduojamąjį tekstą:
$tekstas = "Nurodote koduojamąjį tekstą.";

// Funkcija:
$resultatas = md5($tekstas);

// Kodavimo rezultatas:
print $resultatas;

// Šio tipo kodavimas yra negrįžtamas.
// Jo negalėsite atkoduoti atgal.

?>
```

Rezultatas: įvestas tekstas užkoduotas atrodo taip:

Sled42pssdx6l2022dxyzabc5sd3zdf90edwsx7j2jsalc87ds9s0asdv82ffs0e2saflsa96ax0z1s0xvbj9

Šio algoritmo taikymas užtikrina atsitiktinių simbolių parinkimą ir pateiktos informacijos užkodavimą jais. Turint md5 formatu užkoduotą simbolių rinkinį, jo atkoduoti atgal negalima. Algoritmo naudojimas praktiškai yra neįveikiamas, nes sugeneruota simbolių seka neturi tokios sąsajos su originaliu tekstu.

Svarbi informacija, kuri sistemoje koduojama md5 algoritmu:

- Asmens pažymiai
- Prisijungimo informacija
- Vartotojo talpinama informacija – prisijungimo teisių priskyrimas
- Siunčiamo el.pašto turinio kodavimas

#### 4.2. Greitas rezultatų pateikimas pagal paieškos raktus

Kadangi sistema informaciją saugo duomenų bazėje, tai būtinas greitas informacijos gavimas, atnaujinimas, šalinimas pagal vartotojo pateiktus vykdymo raktus. Koncentruojamasi į priemones mažinančias laiko kaštus. Tad buvo tobulinant sistema kreiptinas dėmesys į šiuos dalykus:

- Informacijos (vartotojų) paieška pagal du parametrus – pavardę, identifikavimo kodą turi užtrukti ne daugiau 3 sekundžių
- Kitais atvejais vartotojas informaciją savo ekrane turi gauti ne vėliau, kaip 5 sekundžių.

Parinkti skaičiai neatsitiktinai, o atsižvelgiant į vartotojo norą laukti.

Paieškos tobulinimo vykdymas:

- Informacijos paieška įvedus ne visus reikalaujamus parametrus – tai leidžia gauti informaciją dalinai žinant informaciją apie ieškomą objektą.

Informacijos šalinimas:

- Pagal vartotojo identifikatorių arba nurodžius kartu pavardę ir vardą
- Senų laiškų pagal datą arba datos intervalą šalinimas – užtikrina vartotojui greita laiškų šalinimą nei kiekvieną pasirenkant atskirai.

#### 4.3. Perėjimo tarp puslapių efektyvinimas

Taikant sistema vis labiau praktiškai randama klaidų, susijusių su perėjimais tarp puslapių. Rastos klaidos fiksuotos ir įvestos nuorodos, vedančios į pagrindinius puslapius. Tai sutaupo vartotojo laiką, didina pačios sistemos kokybę. Panaudojus šią funkciją vartotojai daug greičiau gali pasiekti pagrindinius puslapius.

#### 4.4. El.pašto kodavimas wml programavimo kalba

Mobilūs telefonai geriausiai interpretuoja wml programavimo kalba parašytą kodą, tad vystant sistemos kokybės tobulinimą, naudotas php-wml kodas pakeistas į grynai wml kodą:

```
<wml> <template> <do type="prev"> <prev/> </do> </template>
<card id="InfoCard" title="xh Wap Email Service"> <p align="center">
<big>Wap Email Service</big><br><fieldset title="Email">
Kam : <input type="text" name="toadd" maxlength="32"/>
Nuo : <input type="text" name="fromadd" maxlength="32"/>
Pavadinimas : <input type="text" name="subject" maxlength="32"/>
Turinys: <input type="text" name="body" maxlength="32"/> </fieldset>
<do type="accept" label="Send" name="OK"> <go href="wmlmail.asp" method="post" >
<postfield name="toadd" value="$toadd" /> <postfield name="fromadd" value="$fromadd" />
<postfield name="subject" value="$subject" /> <postfield name="body" value="$body" />
<postfield name="OK" value="OK" />
</go></do></p></card>
</wml>
```

Pasirinkus šį variantą, pagreitėjo laiško išsiuntimas 3 sek, nebeapsitaiko formatų suderinamumo klaidą metančios pranešimo.

## 5. EKSPERIMENTINĖ DALIS

Magistrinio projekto darymo metu buvo sukurta ir KTU gimnazijos serveryje įdiegta m.mokymosi modulių kūrimo PĮ. Prisijungimas prie sistemos galimas adresu:

Prisijungimo vardai: mokinio aplinka: mokytojo aplinka: administratoriaus aplinka:

M.modulį sudaro dvi dalys: SMS ir WAP modulis .Buvo vykdytas šių modulių pasirinktų (dažniausiai naudojamų) funkcijų eksperimentinis tyrimas.

### SMS modulis

Vykdytas išsiunčiamų SMS žinučių eksperimentinis tyrimas. Juo buvo siekiama išsiaiškinti, ar vartotojai, kuriems pasirinkta siųsti gauna žinutes ir kaip greitai jas gauna. Tam tikslui buvo pasirinkti toje pačioje patalpoje esantys ir mobilius telefonus turintys asmenys iš skirtingų mobilaus ryšio operatorių. Toks pasirinkimas leido patikrinti, ar visų tinklų vartotojai gauna žinutes.

Pasirinkimas buvo vykdomas SMS modulio lange pasirinkus nuorodą SMS siuntimas ir atsidariusiame žemiau esančiame lange pasirinkus vartotojų grupę ir įvedus tekstą:



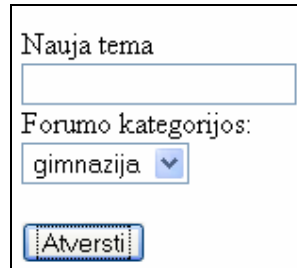
22 pav. SMS siuntimo langas

Išsiųstos žinutės korektiškai buvo patalpintos į duomenų bazę pagal pavardę ir OzekiSMS servisu išsiųstos nurodytiems vartotojams. Visi vartotojai SMS žinutes gavo, tik ne ta tvarka, kokia buvo patalpinta duomenų bazėje, tam įtakos galėjo turėti mobilaus operatorių tinklų apkrova (tyrimas buvo darytas 12 valandą).

## WAP modulis

### *Forumo temos šakos kūrimas*

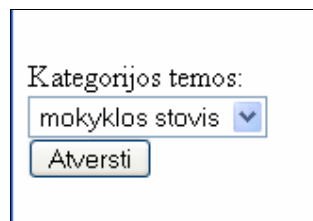
Tyrimo metu buvo siekiama išsiaiškinti, ar vartotojas gali sukurti naują forumo šaką ir joje patalpinti informaciją. Vartotojui rodoma forma ekrane:



Nauja tema  
[input field]  
Forumo kategorijos:  
gimnazija ▼  
Atversti

23 pav. Naujos forumo temos įvedimo langas

Rezultate: vartotojo įvesta nauja tema pavadinimu *mokyklos stovis* forumo kategorijoje *gimnazija*.



Kategorijos temos:  
mokyklos stovis ▼  
Atversti

24 pav. Kategorijos temų pasirinkimo langas

Rezultatas – vartotojas gali savarankiškai įterpti naują forumo kategorijos temą ir vėliau ji gali būti korektiškai pildoma kitų forumo dalyvių.

### *Mokinio registracija pamokai*

Buvo siekiama išsiaiškinti ar mokinys paprastai ir greitai gali prisiregistruoti pamokai savo mobiliu telefonu įvesdamas ir į sistemą nusiųsdamas registravimosi duomenis. Tai leido patikrinti, ar realizuota funkcija korektiškai užtikrina registravimą ar mokiniai nepiktnaudžiauja, registruodami savo pamokoje nesančius draugus. Tuo tikslu registracijai buvo skirta 5 minutės ir registravimasis iš vieno telefono numerio buvo galimas tik vieną kartą.



-Registracija-

Mano slaptažodis:

Kodas:

Jungtis

25 pav. Registracija pamokai forma

Data: 2006-12-05  
Laikas: devyni  
Busena: sėkmingai užregistruotas

26 pav. Sėkmingo prisijungimo pranešimas

Nesėkmingo prisijungimo atveju išvedamas pranešimas – registracija nepavyko. Registracijų skaičius išnaudotas.

#### *Sistemos atsparumas dideliame prisijungiančiųjų vartotojų skaičiui*

Buvo siekiama išsiaiškinti kiek stabiliai dirba sistema prie didelių imituojamų vartotojų skaičiaus. Tuo tikslu buvo vykdomas prisijungimas prie sistemos vartotojų vardus ir slaptažodžius atsitiktine tvarka imant iš duomenų bazės nei įterpant po kelis atsitiktinai sistemos sugeneruotus.

Rezultate matome, kad didėjant prisijungimų skaičiui tuo pačiu laiku (santykinai) didėja laiko trukmė, reikalinga prisijungti prie sistemos. Šie rezultatai gali būti naudingi sistemą pritaikant didesniai vartotojų skaičiui.

## 6. IŠVADOS

1. Atlikta analizė rodo, kad mobilios technologijos veržiasi į mokymo(si) sritį ir išplečia mokymo(si) galimybes, tačiau dėl mobilių technologijų apribojimų kyla papildomų problemų: mažas ekrano dydis, maža įtaiso talpa, tinklo pralaidumas, dėl to reikia nagrinėti mokymo(si) informacijos transformavimą.
2. Remiantis literatūroje aprašytu DRESS metodu, adaptuotas mokymo informacijos transformavimo modelis, kurį sudaro sudaro suformuota žiniatinklio puslapio struktūra, jos padalijimų medis, apskaičiuotos padalijimų reikšmės.
3. Nustatyti tokie automatinio transformavimo žingsniai: ekrano dydžio nustatymas; puslapio informacijos suskaidymas į informacinius blokus; blokų svarbumo nustatymas; atvaizdavimas įvertinant mobilaus įtaiso apribojimus.
4. Informaciniai blokai patalpinti atmintyje pagal dvejtainį medį bei apskaičiuotos jų padalijimų reikšmės, rodančias, kiek tame informacijos bloke esanti informacija svarbi.
5. Nustatytas sąryšis tarp suformuotų informacijos blokų ir mobilaus ekrano skaičiaus – blokuose esančios informacijos taškų skaičius turi būti lygus ekrano taškų skaičiui.

## 7. LITERATŪRA

- [1] C. O'Malley, G. Vavoula, J.P. Glew, J. Taylor, M.Sharples, P. Lefrere, Guidelines for learning/teaching/tutoring in a mobile environment, D4.1 of MOBilearn project at. [Aplankyta 2007 02 17]. Prieinama internete: <http://www.mobilearn.org/download/results/guidelines.pdf>
- [2] Paul Harris, Goin Mobile, in Learning Circuits, American Society for Training and Development. [Aplankyta 2007 04 09]. Prieinama internete: <http://www.learningcircuits.org/2001/jul2001/harris.html>, July 2001.
- [3] A.Čepulkauskas, G.Kulvietis, Mobilių sprendimų kūrimas, VGTU, 2006, Mokomoji medžiaga. [Aplankyta 2007 04 09]. Prieinama internete: [http://leidykla.vgtu.lt/docs/el/Mobiliuju\\_sprendimu\\_kurimas.pdf](http://leidykla.vgtu.lt/docs/el/Mobiliuju_sprendimu_kurimas.pdf)
- [4] Opera Small Screen Rendering. [Aplankyta 2007 03 21]. Prieinama internete: <http://www.opera.com/products/smartphone/smallscreen/>
- [5] Björk, S., Holmquist, L.E., Redström, J., Bretan, I., Danielsson, R., Karlgren, J., Franzén, K.: WEST: A Web Browser for Small Terminals. In: Proceedings of UIST'99, pp. 187–196. Asheville, USA (1999)
- [6] BickMore, T.W., Schilit, B.N.: Digestor: device-independent Access to the world wide web. In: Proceedings of WWW'97, pp. 655–663. Santa Clara, USA (1997), [Aplankyta 2007 03 26]. Prieinama internete: <http://citeseer.ist.psu.edu/bickmore97digestor.html>
- [7] Buyukkokten, O., Garcia-Molina, H., Paepcke, A., Winograd, T.: Power browser: efficient web browsing for PDAs. In: Proceedings of CHI'00, pp. 430–437. Hague, Netherlands (2000). [Aplankyta 2007 04 28]. Prieinama internete: <http://infolab.stanford.edu/~orkut/papers/pb1.pdf>
- [8] Han, R., Perret, V., Naghshineh, M.: WebSplitter: A unified XML framework for multi-device collaborative web browsing. In: Proceedings of CSCW'00, pp. 221–230. Philadelphia, USA (2000), [Aplankyta 2007 05 01]. Prieinama internete: [http://www.cs.colorado.edu/~rhan/CSCI\\_7143\\_002\\_Fall\\_2001/Papers/Han2000\\_WebSplitter.pdf](http://www.cs.colorado.edu/~rhan/CSCI_7143_002_Fall_2001/Papers/Han2000_WebSplitter.pdf)
- [9] Hori, M., Kondoh, G., Ono, K., Hirose, S., Singhal, S.: Annotation-based web content transcoding. Comput. Netw. 33(1–6), 197–211 (2000). [Aplankyta 2007 02 28]. Prieinama internete: <http://www9.org/w9cdrom/169/169.html>
- [10] Badros, G., Borning, A., Marriott, K., Stuckey, P.: Constraint cascading style sheets for the web. In: Proceedings of UIST'99, pp. 73–82. Asheville, USA (1999), [Aplankyta 2007 03 01]. Prieinama internete: <http://www.cs.washington.edu/research/constraints/web/ccss-uwtr.pdf>
- [11] Cohoon, J.P., Paris, W.D.: Genetic placement. IEEE Trans. Comput. Aided Design (6), 956–964 (1987)
- [12] Li-Qun Chen, Dress: A slicing tree based web page representation for various display sizes. [Aplankyta 2007 03 12]. Prieinama internete: <http://www2003.org/cdrom/papers/poster/p260/DRESS-poster/DRESS-poster.htm>
- [13] Chen, Y., Ma, W.Y., Zhang, H.J.: Detecting web page structure for adaptive viewing on small form factor devices. In: Proceedings of WWW'03. Budapest, Hungary (2003)

- [14] Song, R., Liu, H., Wen, J.R., Ma, W.Y.: Learning block importance models for web pages. In: Proceedings of WWW'04. New York, USA (2004). [Aplankyta 2007 03 27]. Prieinama internete: [http://research.microsoft.com/users/jrwen/jrwen\\_files/publications/BlockImportance.PDF](http://research.microsoft.com/users/jrwen/jrwen_files/publications/BlockImportance.PDF)
- [15] Chen, Y., Ma, W.Y., Zhang, H.J.: Detecting web page structure for adaptive viewing on small form factor devices. In: Proceedings of WWW'03. Budapest, Hungary (2003). [Aplankyta 2007 04 12]. Prieinama internete: <http://www2003.org/cdrom/papers/refereed/p297/p297-chen.htm>
- [16] Xing Xie · Chong Wang · Li-Qun Chen · Wei-Ying Ma, An adaptive web page layout structure for small devices. [Aplankyta 2007 04 27]. Prieinama internete : <http://research.microsoft.com/users/chwang/paper/AWPL.pdf>
- [17] Arouna Woukeu, David E. Millard, Feng Tao, Hugh C. Davis, Challenges for Semantic Grid based Mobile Learning. [Aplankyta 2007 04 10]. Prieinama internete: <http://eprints.ecs.soton.ac.uk/11031/>
- [18] Laura Naismith, Peter Lonsdale, Giasemi Vavoula, Mike Sharples, Literature Review in Mobile Technologies and Learning. Report 11 in NESTA FUTURELAB SERIES produced by the University of Birmingham, available at. [Aplankyta 2007 02 24]. Prieinama internete [http://www.nestafuturelab.org/images/downloads/futurelab\\_review\\_11.pdf](http://www.nestafuturelab.org/images/downloads/futurelab_review_11.pdf)
- [19] M-Learning EU Project, m-learning : learning in the palm of your hand. [Aplankyta 2007 02 10]. Prieinama internete: <http://www.m-learning.org/>
- [20] Mobilelearn EU project. [Aplankyta 2007 02 26]. Prieinama internete: <http://www.mobilelearn.org/>
- [21] D. Chu and M. Humphrey, "Mobile OGSINET: Grid Computing on Mobile Devices", 5th IEEE/ACM International Workshop on Grid Computing - RID2004 (associate with Supercomputing 2004). Nov 8 2004, Pittsburgh, PA. [Aplankyta 2007 03 28]. Prieinama internete: <http://www.cs.virginia.edu/~humphrey/papers/MobileOGSI.pdf>
- [22] Millard, D., Woukeu, A., Tao, F. B. and Davis, H. (2005) Experiences with Writing Grid Clients for Mobile devices. In Proceedings of 1st International ELeGI Conference on Advanced Technology for Enhanced Learning, Vico Equense - Napoli (Italy). [Aplankyta 2007 04 21 ]. Prieinama internete: [http://eprints.ecs.soton.ac.uk/10671/01/mobile-grid-clients\\_final-1.doc.pdf](http://eprints.ecs.soton.ac.uk/10671/01/mobile-grid-clients_final-1.doc.pdf)
- [23] IKT taikymas ugdymo procese. [Aplankyta 2007 04 18]. Prieinama internete: <http://www.pedagogika.lt/puslapis/Pranesimai/Konf%202006-02-10%20Buineviciutes.pdf>
- [24] E-learning. [Aplankyta 2007 05 01]. Prieinama internete: [http://isp.webopedia.com/TERM/E/e\\_learning.html](http://isp.webopedia.com/TERM/E/e_learning.html)
- [25] Routledge, Taylor & Francis Group, London, Mobile Learning: A Handbook for Educators and Trainers. [Aplankyta 2007 04 28]. Prieinama internete: <http://iet.open.ac.uk/pp/a.m.kukulskahulme/aboutmobilebook.html>
- [26] Mokomoji medžiaga mokytojų kvalifikacijos tobulinimo programai. [Aplankyta 2007 05 01]. Prieinama internete: [http://209.85.129.104/search?q=cache:9C4VG66THEMJ:www.mkc.lt/dokuments/mokymosi\\_medziaga/Atsiverkime\\_ateiciai.pdf+el-mokymasis&hl=lt&ct=clnk&cd=7&gl=lt](http://209.85.129.104/search?q=cache:9C4VG66THEMJ:www.mkc.lt/dokuments/mokymosi_medziaga/Atsiverkime_ateiciai.pdf+el-mokymasis&hl=lt&ct=clnk&cd=7&gl=lt)

[27] Mike Sharples, The Design of Personal Mobile Technologies for Lifelong Learning.  
[Aplankyta 2007 03 14]. Prieinama internete:  
<http://www.eee.bham.ac.uk/sharplem/Papers/handler%20comped.pdf>

## 8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

Wap – mobilaus interneto protokolas.

Gsm – Globali mobilių sistemų bendravimas.

Gps – vietos nustatymo sistema.

Mms- paveikslukų, filmukų siuntimo paslauga.

Bluetooth and infrared – galimybė sujungti išorinius įtaisus belaidžiu ryšiu.

Sms – trumpoji žinutė, siunčiama iš mobilaus telefono.

GPRS – paketinis duomenų perdavimas.

Specifikacija - sistemos funkcionalumo aprašymas formaliais metodais.

UML – unifikuota modeliavimo kalba (Unified Modeling Language).

RUP - Rational Unifikuotas Procesas (Rational Unified Process).

PA - panaudos atvejai (Use Case).

DB - duomenų bazė

Sistemos gyvavimo ciklas - sistemos plėtros pokyčių nuo koncepcijos iki veikimo pabaigos visuma.

Kompiliatorius - parašytą programą verčiantis į kompiuteriui suprantamus simbolius

Apache server - serveris, leidžiantis vykdyti PHP failus

## 9. PRIEDAI

Priedas 1

Straipsnis pristatytas konferencijoje Informacinės technologijos 2007, 2007 01 31 – 2007 02 01

### **MOBILAUS MOKYMOŠI MODULIAI: KAI KURIE ANALIZĖS IR KŪRIMO ASPEKTAI**

**Darius Bakšys**

*KTU Informatikos fakultetas, Programų inžinerijos katedra, Studentų g. 50, Kaunas*

Straipsnyje supažindinama su SMS ir WAP technologijomis, aprašomos šių paslaugų praktinės ir teorinės pritaikymo galimybės šiandieninėje mokykloje. Pateikiami principai, kuriais remiantis sukurti ir išanalizuoti mobilaus mokymosi moduliai, leidžiantys vartotojui mobiliu įtaisu pasiekti ne tik savo pažymių knygutės, namų darbų užduočių duomenis, bet ir siųsti žinutes į draugų el. pašto dėžutes, vykdyti vartotojų ir sistemos administravimo funkcijas, atlikti užsiregistravimo pamokoje procesą, nusiunčiant savo mobiliu telefonu mokytojo pateiktą kodą. Apibrėžiami nauji pritaikymo būdai, atsakoma į dažniausiai iškylančius klausimus kuriant ir realizuojant mobilaus mokymosi modulius. Pateikiamas praktinio modulių pritaikymo vienoje Kauno švietimo įstaigoje aprašymas su vartotojų vertinimo rezultatais.

#### **IVADAS**

Mobilaus mokymosi ( Mob.mokymosi) modulis – tai programinės įrangos (PI) sudėtinė dalis, įgalinanti informaciją skaityti ir ją siųsti į e-mokymosi portalą mobiliu telefonu ar kitu mobiliu įtaisu (pvz., delniuku). Prisijungimui naudojami mobilūs telefonai, palaikantys SMS ir WAP technologijas. Kuriant sistemas, susiduriama tiek su mobilių įtaisų, tiek su sistemų suderinamumo sunkumais. Pagrindinės problemos, susijusios su mobiliais įtaisais – telefonais yra šios: mažas ekranas, lėtas internetinis ryšys, maža telefono vidinė atmintis. Šios problemos šalinamos diegiant naujos (trečios – 3G) kartos tinklus užtikrinančius greitesnį ir patikimesnį ryšį, mobilūs telefonai kuriami kartu su fotokameromis ir kt. priedais, praplečiančiais jų pritaikymo galimybes. Vidinė atmintis didinama pritaikant naujausius techninius sprendimus – pasiekimus.

Mokymo įstaigos kaip įstaigos, kurios moko jaunąją kartą po truputį ima naudoti mobiliausias technologijas savo mokymo ir administravimo procese. Vis labiau suprantama, kad savęs pristatymas ir pasiekimas iš bet kurios vietos ir bet kuriuo paros metu didina vartotojų skaičių, kas leidžia būti matomam, labiau domėtis mokykla, prisidėti prie mokymosi ir informavimo proceso gerinimo, o tuo pačiu ir savo išprusimo lygio kėlimo.

#### **MOB. MOKYMOŠI MODULIŲ KŪRIMO PROCESAS**

Procesas susideda iš modulių lygių išskyrimo, jų detalizavimo bei praktinio pritaikymo aprašymo, parodant sukurtų modulių naudą mokymo ir administravimo procese.

Modulio sudėtinės dalys

M.mokymosi modulio lygiai:

Vartotojo
Procesų
Architektūros
Realizacijos

**21 pav. M.mokymosi modulio lygiai.**

- Vartotojo lygis

Skiriamos šios modulio vartotojų grupės – mokyklos bendruomenės nariai:

**1 lentelė. Modulio vartotojai.**

Administracija	Asmenys, vykdančys ir prižiūrintys mokymosi ir valdymo procesus
Mokytojai	Asmenys, atsakingi už kokybišką mokslo pateikimą
Mokiniai	Mokymosi proceso dalyviai, įsisavinantys mokymosi medžiagą
Tėvai (globėjai)	Mokinių tėvai (globėjai), prisidedantys prie mokymosi kokybės erinimo.
Svečiai	M-mokymosi modulio lankytojai, galintys susipažinti su m-internete pristatoma mokykla, jos tikslais, pasiekimais ir pan.

Moduliu gali naudotis visi mokyklos bendruomenės nariai, turintys mobilius telefonus, palaikančius SMS ir WAP technologijas (telefonai, sukurti prieš 2-3 metus). SMS technologija palaiko visi mobilūs telefonai. Mobiliam telefone turi būti aktyvuota galimybė naudotis mobiliu internetu (aktyvaciją įvykdo mobilaus telefono operatorius išsakant savo pageidavimą).

- Procesų lygis

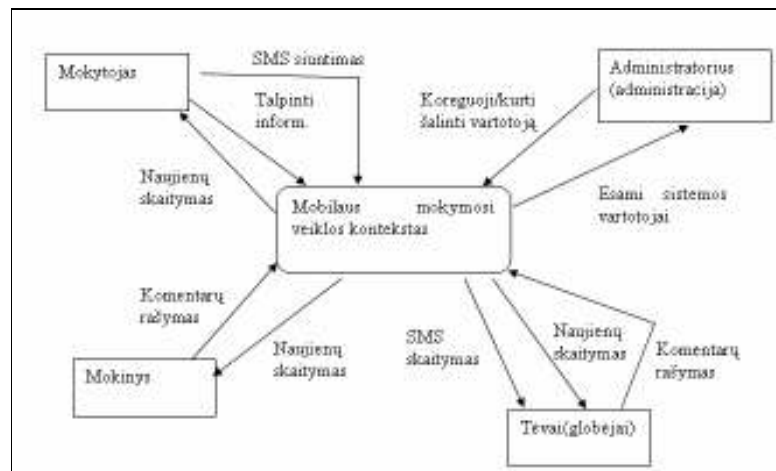
Šiame lygyje nusakomi ir paaiškinami modulio procesai:

**2 lentelė. Procesai ir jų paaiškinimai.**

Procesas	Paaškinimas
Informacijos talpinimas diskusijų lentoje	Dalyvavimas vartotojų diskusijose (komentarų rašymas, skaitymas) prisijungus mobiliu telefonu prie diskusijų lentos
Skelbimo talpinimas (skaitymas) skelbimų lentoje	Skelbimo talpinimas (skaitymas) mobiliu telefonu prisijungus prie skelbimų lentos
Savo pažymių peržiūra	Gautų pažymių peržiūra prisijungus prie savo darbo aplinkos mobiliu telefonu
Pamokų tvarkaraščio talpinimas (peržiūra)	Pamokų tvarkaraščio įkėlimas bei peržiūra prisijungus mobiliu telefonu prie mokyklos duomenų bazės
Registravimasis pamokai	Galimybė naudojantis WAP užsiregistruoti pamokai nusiunčiant mobiliu telefonu mokytojo pateiktą pamokos kodą.
Namų darbų užduočių gavimas	Namų darbų užduočių talpinimas į mokyklos duomenų bazę
Savarankiškų darbų gavimas	Medžiagos skirtos pasiruošti kontroliniams, testams talpinimas į mokyklos duomenų bazę prisijungus mobiliu telefonu

Vartotojų administravimas	Vartotojų kūrimas, šalinimas, naujinimas prie sistemos prisijungus mobiliu telefonu
Žinutės siuntimas į el.pašto dėžutę	El.žinutės siuntimas į pasirinkto abonento el.pašto dėžutę WAP pagalba
Laisvalaikio medžiagos (muzikos, piešinių) parsisiuntimas	.mid ir .jpg formato informacijos parsisiuntimas laisvalaikio tikslais iš mokyklos serverio.
SMS siuntimas	SMS siuntimas iš mobilaus telefono, prijungto prie mokyklos serverio
SMS gavimas	SMS gavimas ir skaitymas savo darbo kompiuteryje

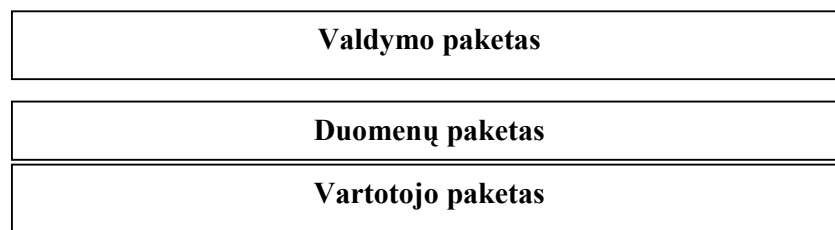
Vartotojo ir proceso lygius sujungia konteksto diagrama



22 pav. Konteksto diagrama.

- Architektūros lygis

Naudojamas 3 lygių architektūros modelis:

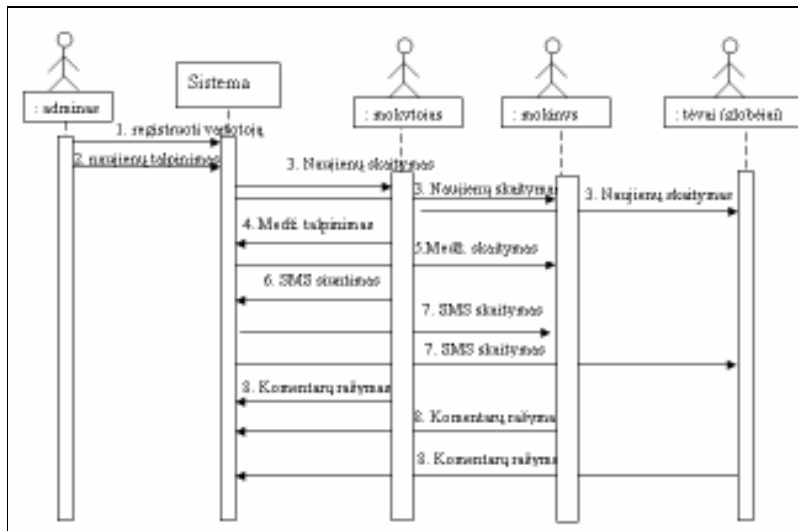


23 pav. Trijų lygių architektūros modelis.

Kiekvienas paketas atitinka veiklos tipą:

- Duomenų paketas – duomenų struktūrų projektavimas.
- Valdymo paketas – programos (sistemos funkcionalumo) projektavimas. Veiklos paketą sudaro valdančios klasės, kuriomis įgyvendinamos veiklos taisyklės
- Vartotojo paketas, kurį sudaro formos (ribinės klasės), kurių pagalba sistemos vartotojai bendrauja su sistema. Šiame sluoksnyje yra prisijungimo forma, sistemos administravimo formos.





24 pav. Sekų diagrama.

- Realizacijos lygis

WAP modulis kuriamas naudojant specifinę, mobilaus interneto kūrimo puslapių programavimo kalbą – wml. Ji greitai perprantama. Tačiau tenka šią programavimo kalbą derinti su PHP kalba, nes WML kalba neįmanoma paimti duomenų iš duomenų bazės, tai sėkmingai atliekama WML kode įterpiant PHP kodą bei nurodant simbolių koduotę baziniu žodžiu – charset.

WAP ir SMS moduliai



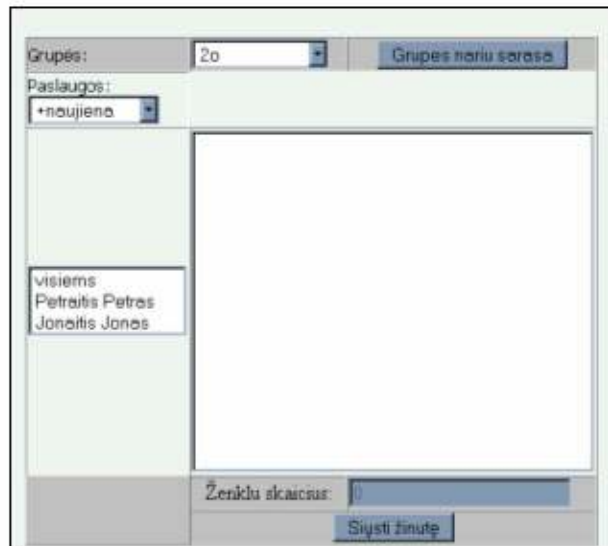
25 pav. WML vartotojo sąsaja.

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.3//EN" "http://www.wapforum.org/DTD/wml13.dtd">
<wml>
  <card id="card1" title="WML Tutorial">
    <p>Hello World</p>
  </card>
</wml>
  
```

26 pav. WML kalbos kodas.

SMS modulis – programuojamas naudojant PHP programavimo kalbą, žinučių siuntimui ir nuskaitymui iš duomenų bazės galima naudoti Ozeki SMS server ([www.ozeki.hu](http://www.ozeki.hu)) nurodant duomenų bazės pavadinimą.



27 pav. SMS modulio vartotojo darbo aplinka.

### 3. PRAKTINIO MODULIŲ PRITAIKYMO VIENOJE KAUNO MOKYKLOJE TRUMPAS APRAŠAS

Mokykloje įdiegti sukurtas WAP ir SMS moduliai, jų pagalba mokyklos bendruomenės nariai gali naudotis aukščiau aprašytais funkcijomis. Kiekvienas pageidaujantis iš sistemos administratoriaus gauna prisijungimo prie sistemos duomenis ir gali naudotis sistemos galimybėmis. Kiekvienas vartotojas prisijungęs dirba savo darbo aplinkoje, gali pasiekti tik jam, kaip vartotojui priskirtą informacijos dalį. Vartotojas už mobilaus interneto paslaugas (WAP) susimoka pats, už SMS siuntimą per mokyklos serverį, jam mokėti nereikia. Išbandyti WAP modulį galima mobiliu telefonu prisijungus adresu: <http://www.kovo11.kaunas.lm.lt/wap/index.wml>. Peržiūrai internetu rekomenduojama parsisiųsti PĮ: Mozilla Firefox adresu: <http://www.mozilla.com/en-US/firefox/all.html> bei įskiepi (wmlbrowser), adresu: <http://wmlbrowser.mozdev.org/installation/wmlbrowser.html>. Prisijungimo slaptažodis: lv04.

Vartotojų vertinimo rezultatai:

Buvo apklausti mokyklos bendruomenės nariai. Pateikti klausimai susiję su SMS ir WAP panaudojimu. 65% pritaria mokyklos WAP svetainės sukūrimui., 75% noriai naudojami SMS siuntimo iš mokyklos serverio paslaugomis. 60% vartotojų pageidauja susitikimo, kuriame plačiau būtų papasakota apie WAP ir SMS platesnį pritaikymą mokymo ir administravimo procese mokymo įstaigose.

### LITERATŪROS SĄRAŠAS

- [1] **H. Singh.** Leveraging Mobile and Wireless Internet. Prieiga internetu: <http://www.learningcircuits.org/fundamentals.html>

- [2] From e-learning to m-learning, chapter 4, Mlearning Initiatives in 2001. Prieiga internetu: [http://learning.ericsson.net/mlearning2/project\\_one/thebook/chapter4.html#telenor](http://learning.ericsson.net/mlearning2/project_one/thebook/chapter4.html#telenor)
- [3] Resources. Prieiga internetu: <http://www.m-learning.org/resources.shtml>

Mobile learning module: somethink aspects of creating and analysing.

In this article is speaking about using of SMS and WAP technology in school. Analyse m.learning module, which for user let connect to homework data, send message to e.mail inbox, execute function of users and system administration. Define new methods, answer to question about m.learning module. Speaking about practical module use in one Kaunas school.

## Priedas 2

Testavimo rezultatų lentelė

Testo nr.	Aprašas	Kas padaryta	Kas gauta
1.	Prisijungimo testavimas ..... .....	Įvestas duomenų bazėje nesantis prisijungimo slaptažodis	Pranesimas - neprisijungta

## Priedas 3

Išsiūtų SMS žinučių laiko vertinimas:

Vartotojas / Operatorius	1	2	3	4	5
Tele2	Iki 15 s		Iki 20 s	Iki 15 s	
Omnitel		Iki 10 s			
Bite					Iki 5 s

## Priedas 4

Sistemos atsparumas dideliems vartotojų skaičiams

Prisijungimų kiekis	10	20	30	40	50
Laikas (sek)	3	7	10	12	15